LIBRARY OF THE
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

621.31921
Al54p
1975

[redacted]

MC

21.31921
Al54p
1975

ENGINEERING LIBRARY
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS

Proceedings

THIRTEENTH ANNUAL ALLERTON CONFERENCE ON CIRCUIT AND SYSTEM THEORY

October 1-3, 1975

Allerton House, Monticello, Illinois
Sponsored by the
Department of Electrical Engineering and the
Coordinated Science Laboratory of the
University of Illinois at Urbana-Champaign

ENGINEERING LIBRARY

10e

PROCEEDINGS

THIRTEENTH ANNUAL ALLERTON CONFERENCE ON CIRCUIT AND SYSTEM THEORY

E. S. Davidson
T. N. Trick
Conference Co-Chairmen

Conference held

October 1, 2, 3, 1975
Allerton House
Monticello, Illinois

Sponsored by

The Department of Electrical Engineering
and
The Coordinated Science Laboratory
of the
UNIVERSITY OF ILLINOIS
at
Urbana-Champaign

621.3192l
Al54p
1975

ENGINEERING LIBRARY

FOREWORD

The Thirteenth Annual Allerton Conference on Circuit and System Theory was held at the Allerton House on October 1-3, 1975. The Conference was sponsored by the Department of Electrical Engineering and the Coordinated Science Laboratory of the University of Illinois at Urbana-Champaign. These PROCEEDINGS contain the papers presented at the Conference.

In addition to the large number of submitted papers, there were several organized sessions of invited papers. These were the sessions on: Current Research on Advanced Automation organized by G. N. Saridis, Source Coding organized by M. B. Pursley and L. D. Davisson, Multiple-User Communication and Coding organized by J. L. Massey, and Digital Communications organized by R. Lugannani. We are grateful to these organizers for their time and effort on behalf of the Conference, and we thank the authors in these sessions for their excellent contributions. We are also grateful to the other session chairmen and authors for their contributions to the success of the Allerton Conference.

Dr. R. W. Lucky delivered the plenary address entitled "Communication Systems, LSI, and Microprocessors" at the joint meeting with the IEEE Group on Information Theory chaired by R. T. Chien. We are indebted to Dr. Lucky for his timely and stimulating presentation.

The support and encouragement of Professor E. C. Jordan, Head of the Department of Electrical Engineering and Professor R. T. Chien, Director of the Coordinated Science Laboratory are appreciated and acknowledged. We thank our colleagues for their generous assistance in organizing the Conference: J. Abraham, D. Bertsekas, J. B. Cruz, Jr., R. A. Flower, A. H. Haddad, P. V. Kokotović, H. F. Li, W. Mayeda, G. Metze, W. R. Perkins, F. P. Preparata, M. B. Pursley, D. V. Sarwate, M. E. Van Valkenburg, D. L. Waltz, and N. Wax.

We are grateful to Mr. Harold Lawler for his handling of the business aspects of the Conference. We are also grateful to Mrs. Rosa Townsend for her help in the preparation of the Conference. Thanks are also due to Mr. W. Kao for coordinating the projectionists and the visual aid equipment. Special thanks are due to Ms. Marge Sheehan for her expert and skillful coordination of the Conference arrangements.

E. S. Davidson
T. N. Trick

TABLE OF CONTENTS

Foreword iii

I-A: INFORMATION THEORY AND CODING
Chairman: W. L. Root, University of Michigan

MULTIPLE STACK SEQUENTIAL DECODING ALGORITHM . . . 1
P. R. Chevillat and D. J. Costello, Jr., Illinois Institute of Technology, Chicago

NOISE MEASUREMENTS AND GOOD CODES FOR MULTI-PATH TREE ENCODED SPEECH 2
C.-W. Law, J. B. Anderson and R. B. Cirjanic, McMaster University, Canada

THE CAPACITY REGION FOR GENERAL BROADCAST CHANNELS 11
W. J. Leighton and H. H. Tan, Princeton University

THE USE OF COMBINATION SEQUENCES IN A MULTIPLE ACCESS ENVIRONMENT 21
L. B. Milstein, Rensselaer Polytechnic Institute, New York

BIDIRECTIONAL CONVOLUTIONAL CODES 28
A. Elahi-Taleghani and D. J. Costello, Jr., Illinois Institute of Technology, Chicago

I-B: SYSTEM REPRESENTATION AND REDUCTION
Chairman: Lt. Col. W. J. Rabe, Air Force Office of Scientific Research

ROUTH APPROXIMATION METHOD AND SINGULAR PERTURBATIONS 29
M. F. Hutton, The Singer Company, Little Falls, New Jersey

PROJECTIVE METHODS FOR REDUCING LINEAR DYNAMIC SYSTEMS AND APPLICATIONS TO ESTIMATION AND CONTROL 39
J. Hickin and N. K. Sinha, McMaster University, Canada

TWO STAGE DESIGN OF SINGULARLY PERTURBED LINEAR REGULATORS 48
J. H. Chow, University of Illinois

AN ITERATIVE ALGORITHM FOR DETERMINING A TRANSFER FUNCTION FROM AMPLITUDE FREQUENCY RESPONSE DATA . . 58
M. T. Jong and K. S. Shanmugam, Wichita State University, Kansas

TIME DOMAIN APPROXIMATION USING A DAMPED LEAST-SQUARES ALGORITHM 59
W. S. Ishak and N. K. Sinha, McMaster University, Canada

I-C: COMPUTER SYSTEM MODELLING
Chairman: R. Flower, University of Illinois

BOUNDS AND BOUNDING FUNCTIONS FOR A STATIONARY GI/G/1 QUEUE 65
S. Calo, RCA David Sarnoff Res. Center, Princeton, New Jersey and S. Schwartz, Princeton University

LOOP DATA COMMUNICATION WITH EXTRACTABLE ACKNOWLEDGMENT SIGNALS 75
J. J. Metzner, Wayne State University

ANALYSIS OF REAL TIME SYSTEMS: AN INFORMATION THEORETIC APPROACH 85
W. R. Adrion, P. A. Frick, and P. A. Szulewski, Oregon State University

FAIL RESISTANT MULTIPROCESSING NETWORKS 93
G. G. L. Meyer, The Johns Hopkins University

ERROR ANALYSIS IN A MULTIPLE PROCESSOR SOFTWARE SYSTEM 103
P. D. Shea and J. W. Austin, TRW Systems Corp., Redondo Beach, California

THE INFLUENCE OF COMPUTER HARDWARE AND COMMUNICATION LINES ON THE COST OF DISTRIBUTED COMPUTER SYSTEMS 104
L. Ambrosini Guaccimanni and G. Bucci, University of Bologna, Italy

II-A: CURRENT RESEARCH ON ADVANCED AUTOMATION
Organizer and Chairman: G. N. Saridis
Purdue University

A MINICOMPUTER FACILITY FOR ADVANCED AUTOMATION RESEARCH 114
K. S. Fu and E. Persoon, Purdue University

SOME ALTERNATIVE FORMULATIONS OF MANIPULATOR DYNAMICS FOR COMPUTER SIMULATION STUDIES 124
H. Hemami, V. C. Jaswa, and R. B. McGhee, The Ohio State University

THREE DIMENSIONAL PATTERN RECOGNITION FOR THE ROBOTIC SYSTEM 141
J. Y. S. Luh and E. S. Yam, Purdue University

NEW PROGRESS IN ROBOTICS 151
D. L. Waltz, University of Illinois

USING A MICROCOMPUTER TO CONTROL A SMALL ROBOTIC CART 158
R. A. Northouse, University of Wisconsin-Milwaukee

II-B: OPTIMIZATION METHODS
Chairman: D. Bertsekas, University of Illinois

OPTIMIZATION WITH NONSTANDARD COST FUNCTIONALS . . 168
M. B. Subrahmanyam and E. D. Eyman, University of Iowa

APPLICATION OF MATHEMATICAL PROGRAMMING TO OPTIMAL CONTROL PROBLEMS WITH UNORTHODOX COST FUNCTIONALS AND CONSTRAINTS: A SUMMARY OF RESULTS 174
D. P. Sudhakar and E. D. Eyman, University of Iowa

NEW RESULTS IN THE LEAST PTH APPROACH TO MINIMAX DESIGN 184
J. W. Bandler, McMaster University, Canada, C. Charalambous, University of Waterloo, Canada and J. H. K. Chen and W. Y. Chu, Bell-Northern Research, Ottawa, Canada

A GENERAL METHOD FOR APPROXIMATION BASED ON THE METHOD OF MULTIPLIERS 192
D. P. Bertsekas, University of Illinois

II-C: ESTIMATION, DETECTION AND FILTERING
Chairman: J. B. Thomas, Princeton University

A MODIFIED SIGN DETECTOR BASED ON A FOUR-LEVEL INPUT QUANTIZATION 204
S. A. Kassam, University of Pennsylvania and J. B. Thomas, Princeton University

LOCALLY OPTIMUM RECEIVER STRUCTURES FOR KNOWN SIGNALS IN NON-GAUSSIAN NARROWBAND NOISE 211
J. W. Modestino, Rensselaer Polytechnic Institute, New York

ESTIMATORS SUPERIOR TO THE ML IN MULTIVARIATE NORMAL MODELS 220
P. Papantoni-Kazakos, Rice University

ON A RECEIVER FOR CHANNELS WITH MEMORY 227
P. K. Varshney and A. H. Haddad, University of Illinois

LIKELIHOOD FUNCTIONALS AND CHERNOFF ERROR PROBABILITY BOUNDS FOR DISCRIMINATING BETWEEN MARKOV PROCESSES 234
B. W. Stuck, Bell Laboratories, Murray Hill, New Jersey and C. M. Newman, Indiana University

HOW TO TRACK A SWARM OF FIREFLIES BY OBSERVING THEIR FLASHES 240
D. L. Snyder and P. M. Fishman, Washington University School of Medicine, St. Louis, Missouri

AN APPROXIMATION FOR THE AMPLITUDE DISTRIBUTION OF SHOT NOISE 241
O. Yue, R. Lugannani and S. O. Rice, University of California, San Diego

II-D: NETWORK THEORY

Chairman: M. E. Van Valkenburg, University of Illinois

GENERAL PROCEDURE FOR MULTIVARIABLE POLYNOMIAL POSITIVITY AND NONNEGATIVITY TEST WITH APPLICATIONS 251
N. K. Bose and A. R. Modarressi, University of Pittsburgh

PULSE TO IMPULSE COMPRESSION VIA DISTRIBUTED FEEDBACK 258
D. Hazony, Case Western Reserve University

ON THE ANALYSIS AND SYNTHESIS OF A CLASS OF NONLINEAR CIRCUITS 261
M. Lal and K. Sam Shanmugam, Wichita State University

A SEQUENTIAL METHOD OF SOURCE TRANSFORMATIONS . . 269
S. Shinoda, Chuo University, Japan, T. Maruyama, Science and Technology Agency, Tokyo, Japan and Y. Kajitani, Tokyo Institute of Technology, Japan

THE DISTRIBUTION OF THICK-FILM RESISTOR VALUES . . 277
A. R. Thorbjornsen, The University of Toledo

EXPERIMENTAL RESULTS WITH LINEAR CONTROLLED RESISTANCE AND CONDUCTANCE 284
N. R. Malik and G. L. Jackson, The University of Iowa

DISTRIBUTED PARAMETER SENSITIVITY: O'SHEA, WYNDRUM AND A NEW DP SYNTHESIS METHOD 297
J. K. Linn, Sandia Laboratories, Albuquerque, New Mexico and S. Karni, The University of New Mexico

III-A: SOURCE CODING

Organizer: M. B. Pursley, University of Illinois
Chairman: L. D. Davisson, University of Southern California

TREE CODING OF STATIONARY BLOCK ERGODIC SOURCES WITH A FIDELITY CRITERION 307
H. H. Tan, Princeton University

ENCODING OF A CONTINUOUS SOURCE BY TWISTED MODULATION 314
K. Yao, University of California, Los Angeles

OPTIMAL SOURCE CODING FOR STATIONARY NONERGODIC SOURCES 325
D. L. Neuhoff, The University of Michigan

IMAGE ENCODING WITH A FIDELITY CRITERION 335
B. M. Leiner, Georgia Institute of Technology

ADAPTIVE DATA-COMPRESSION FOR COMMUNICATION NETWORKS 345
H. Nussbaum and Izhak Rubin, University of California, Los Angeles

III-B: SYSTEM THEORY

Chairman: I. B. Rhodes, Washington University, St. Louis, Missouri

OBSERVERS OF LINEAR FUNCTIONALS 356
B. W. Dickinson, Princeton University

ON THE IRREDUCIBLE JORDAN CANONICAL FORM REALIZATION 363
Y. Funahashi, Nagoya University, Japan

NEW PROBLEMS IN PARTIAL REALIZATION THEORY . . . 373
J. R. Roman, The Analytic Sciences Corporation, Reading, Massachusetts

ON TIME DOMAIN SENSITIVITY OF LINEAR SYSTEMS . . 382
M. Eslami and D. P. Brown, University of Wisconsin-Madison

SEQUENCE GENERATION BY AUTONOMOUS DISCRETE-TIME SYSTEMS: FURTHER RESULTS 388
M. E. Kaliski, Northeastern University and
Q. L. Klein, Signal Processing Systems, Inc., Waltham, Massachusetts

EXPLICIT CHARACTERIZATION OF THE STABILIZABILITY REGION OF LINEAR TIME-INVARIANT SYSTEMS 397
F. D. Galiana, University of Michigan

III-C: GRAPH THEORY
Chairman: S. L. Hakimi, Northwestern University

A GRAPH ALGORITHM FOR THE OPTIMAL TEARING OF A SPARSE MATRIX 407
A. Sangiovanni-Vincentelli, University of California, Berkeley

GENERATION OF PATHS BY ELEMENTARY TREE TRANSFORMATIONS 414
S. Shinoda, Chuo University, Japan and W. Mayeda, University of Illinois

SYSTEMATIZATION OF A NEW METHOD FOR GENERATION OF ALL PATHS 421
J. E. Rubinstein, Instituto Costarricense de Electricidad, San José, Costa Rica

A SHORTEST PATH CALCULATION PROGRAM BASED ON CODE GENERATION TECHNIQUE 431
S. Goto, T. Ohtsuki and T. Yoshimura, Nippon Electric Co., Ltd., Japan

EMBEDDING AND EXISTENCE PROBLEMS IN TREES 440
A. N. Patrinos, International Business Machines, Greece and S. L. Hakimi, Northwestern University

A BACKTRACKING ALGORITHM FOR TESTING VERTEX-INDUCED SUBGRAPHS 450
J. D. Johannes, The University of Alabama, Huntsville, D. C. Schmidt, Vanderbilt University and L. E. Druffel, USAF Academy

III-D: SIGNAL PROCESSING AND FILTERING
Chairman: M. Lal, Wichita State University

AMPLITUDE FILTERING USING A TIME VARIABLE TRANSFORMATION 457
E. Singer and A. V. Oppenheim, Massachusetts Institute of Technology

RECURSIVE IMAGE ENHANCEMENT WITH A NEAREST NEIGHBORS IMAGE SENSOR MODEL 467
J. C. Stamm and R. Priemer, University of Illinois at Chicago Circle

A LINEAR PHASE DIGITAL DATA FILTER WITH PREDICTABLE EYE OPENING 477
H. C. Chan and C. R. Carter, McMaster University, Canada

TWO DESIGN METHODS FOR MINIMUM-PHASE FIR DIGITAL FILTERS 485
F. E. Burris, Bradley University

SYNTHESIS OF EQUIRIPPLE RECURSIVE TWO-DIMENSIONAL DIGITAL FILTER 495
H. S. Sekhon, Punjab Agricultural University, India and V. P. Sinha, I.I.T., Kanpur, India

A MULTIRATE DIGITAL FILTER DESIGN METHOD USING CHEBYSHEV APPROXIMATION 505
A. R. Butz, Northwestern University

OPTIMIZATION OF TIME LIMITED OUTPUTS IN BAND LIMITED CHANNELS 515
P. Dines and D. Hazony, Case Western Reserve University

IV-A: MULTIPLE-USER COMMUNICATION AND CODING
Organizer and Chairman: J. L. Massey, University of Notre Dame

COMMUNICATION NETWORKS UNDER DELAY AND CONNECTIVITY CONSTRAINTS 523
J. Hartman and I. Rubin, University of California, Los Angeles

CODING THEOREMS FOR SUPERPOSABLE CODES 530
J. B. Anderson, McMaster University, Canada

SUB-BAUD CODING 539
J. L. Massey and J. J. Uhran, University of Notre Dame

CODING FOR MULTIPLE-ACCESS COMMUNICATIONS SATELLITES 548
W. W. Wu, COMSAT Laboratories, Clarksburg, Maryland

IV-B: STABILITY THEORY AND NONLINEAR SYSTEMS
Chairman: G. Zames, McGill University, Canada

ON THE STABILIZATION OF NONLINEAR SYSTEMS BY DITHER 560
N. Shneydor, Technion, Haifa, Israel and G. Zames, McGill University, Canada

L_2-STABILITY AND INSTABILITY OF LARGE SCALE SYSTEMS DESCRIBED BY INTEGRODIFFERENTIAL EQUATIONS 569
R. K. Miller and A. N. Michel, Iowa State University

STABILITY OF MULTIVARIABLE FEEDBACK SYSTEMS CONTAINING ELEMENTS WHICH ARE OPEN-LOOP UNSTABLE 580
R. K. Miller and A. N. Michel, Iowa State University

ANALYSIS OF A CLASS OF NONLINEAR SYSTEMS WITH A MULTIPLICATIVE FEEDBACK 590
R. A. Curtis, Rome Air Development Center, Rome, New York and T. N. Lee, The George Washington University

STABILITY OF A POWER SYSTEM - THE EFFECT OF GOVERNOR CONTROL ACTION 598
L. Jenkins and R. Liu, University of Notre Dame

AMBIGUOUS BEHAVIOR OF LOGIC BISTABLE SYSTEMS . . . 605
M. Hurtado and D. L. Elliott, Washington University, St. Louis, Missouri

PRACTICAL SENSITIVITY REDUCTION FOR LINEAR AND NONLINEAR SYSTEMS 612
R. M. De Santis and J. Conan, Ecole Polytechnique, Canada

IV-C: COMPUTATIONAL METHODS FOR CIRCUIT ANALYSIS AND DESIGN
Chairman: J. Bandler, McMaster University, Canada

A NEW ALGORITHM FOR ANALYSIS OF NONLINEAR RESISTIVE NETWORKS 622
D. P. Bertsekas, University of Illinois

TOLOPT - A PROGRAM FOR OPTIMAL, CONTINUOUS OR DISCRETE, DESIGN CENTERING AND TOLERANCING 633
J. W. Bandler, McMaster University, Canada, J. H. K. Chen, Bell-Northern Research, Ottawa, Canada, P. Dalsgaard, Aalborg University, Denmark and P. C. Liu, Bell-Northern Research, Verdun, Canada

EFFICIENT CALCULATION OF EXACT GROUP DELAY SENSITIVITIES 643
J. W. Bandler and M. R. M. Rizk, McMaster University, Canada and H. Tromp, University of Ghent, Belgium

ELEMENT COMBINATIONS FOR SYSTEMATIC MACRO-MODELLING OF INTEGRATED CIRCUITS 653
L. C. McAfee, Jr., The University of Michigan

COMPUTER-AIDED DESIGN OF BROAD-BAND MICROWAVE CLASS-C TRANSISTOR AMPLIFIERS 663
F. Filicori and V. A. Monaco, Università di Bologna, Italy

SENSITIVITY CALCULATIONS WITH PHANTOM SOURCES . . 672
M. H. Miller, The University of Michigan-Dearborn

ON SOLVABILITY AND SOLUTION OF LINEAR ACTIVE NETWORKS 681
M. H. Miller, The University of Michigan-Dearborn

IV-D: DIGITAL SYSTEMS
Chairman: F. Hohn, University of Illinois

SIMPLIFIED MULTIPLE STUCK-AT-FAULT TEST GENERATION TECHNIQUES 682
C. D. Latino and J. G. Bredeson, The Pennsylvania State University

ON THE EXISTENCE OF COMBINATIONAL NETWORKS WITH ARBITRARY MULTIPLE REDUNDANCIES 692
J. E. Smith and G. Metze, University of Illinois

GENERAL DESIGN RULES FOR THE CONSTRUCTION OF m-OUT-OF-n TOTALLY SELF-CHECKING CHECKERS 704
J. E. Smith and G. Metze, University of Illinois

A PSEUDOBOOLEAN MULTIVALUED SWITCHING ALGEBRA . . 716
J. Dussault and M. Krieger, University of Ottawa, Canada

MULTI-VALUED ASYNCHRONOUS MODULES 727
A. S. Wojcik, Illinois Institute of Technology, Chicago

DESIGN OF TOTALLY SELF-CHECKING ASYNCHRONOUS SEQUENTIAL MACHINES 737
F. Ozguner, I.B.M., T. J. Watson Research Center and G. Metze, University of Illinois

DIAGNOSIS OF FAULTS IN LINEAR TREE NETWORKS . . . 747
S. C. Seth, University of Nebraska and K. L. Kodandapani, Indian Institute of Technology, Kanpur, India

V-A: DIGITAL COMMUNICATIONS

Organizer and Chairman: R. Lugannani,
University of California, San Diego

PARAMETER ESTIMATION FOR DIGITAL DATA SIGNALS . . 756
L. E. Franks, University of Massachusetts

ON OPTIMAL PULSE-SHAPE DESIGN FOR DATA TRANSMISSION 764
F. S. Hill, Jr., University of Massachusetts

ROBUST QUANTIZATION WITH GENERALIZED MOMENT CONSTRAINTS 774
V. D. Vandelinde, The Johns Hopkins University

SELECTED TOPICS IN HIGH-SPEED DATA TRANSMISSION . 781
R. D. Gitlin, Bell Laboratories, Holmdel, New Jersey

UPPER AND LOWER BOUNDS FOR ERROR PROBABILITY IN PULSE COMMUNICATION SYSTEMS 782
P. J. McLane, Queen's University, Canada

DIGITAL SYSTEM PERFORMANCE IN IMPULSIVE NOISE ENVIRONMENT: A SURVEY 792
T. H. Huynh and M. Lecours, Laval University, Canada

V-B: CONTROL SYSTEMS

Chairman: J. S. Meditch, University of
California, Irvine

A MODIFIED CONFIGURATION FOR LINEAR MULTIVARIABLE SERVOMECHANISMS 801
C. A. Wolfe, Rockwell International, Anaheim, California and J. S. Meditch, University of California, Irvine

OPTIMAL CONTROL OF DELAY SYSTEMS 809
Z. Bien and D. H. Chyung, The University of Iowa

ON THE IMBEDDED SOLUTION OF LINEAR SINGULAR CONTROL PROBLEM WITH DELAY 813
M. Jamshidi and M. Razzaghi, Pahlavi University, Shiraz, Iran

OPTIMAL STAIRCASE INPUT SIGNAL FOR SYSTEM IDENTIFICATION 829
B. Kuszta, Warsaw Technical University, Poland and N. K. Sinha, McMaster University, Canada

A RECURSIVE ALGORITHM FOR TIME-OPTIMAL CONTROL OF SINGULARLY PERTURBED SYSTEMS 839
S. H. Javid, University of Illinois

OPTIMAL STRATEGIES IN SIMPLIFIED TWO-PLAYER POKER 848
D. J. Halski and I. B. Rhodes, Washington University, St. Louis, Missouri

FREQUENCY-DOMAIN DESIGN OF FEEDBACK SYSTEMS FOR SPECIFIED INSENSITIVITY OF TIME-DOMAIN RESPONSE TO PARAMETER-VARIATION 859
K. R. Krishnan, Clarkson College of Technology, Potsdam, New York and A. Cruickshanks, Raytheon Company, Bedford, Massachusetts

V-C: COMPUTER DESIGN AND UTILIZATION
Chairman: A. Wojcik, Illinois Institute of Technology, Chicago

NEAR OPTIMAL ALGORITHM FOR THE PARALLEL EVALUATION OF LINEAR RECURRENCES 867
A. Barak and M. Snir, The Hebrew University of Jerusalem, Israel

TWO METHODS FOR IMPROVING THROUGHPUT OF A PIPELINE 873
W. Mayeda, University of Illinois

ASYNCHRONOUS COORDINATION OF DATA OPERATIONS: A SCHEME EXPLOITING OVERLAP POTENTIAL 887
M. J. Stucki and M. L. Pepper, Washington University, St. Louis, Missouri

SYNTHESIS OF LEVEL SEQUENTIAL CIRCUITS: FURTHER DEVELOPMENT OF A PROCEDURE BASED ON A PETRI-NET TYPE OF BEHAVIORAL DESCRIPTION 896
M. J. Stucki, Washington University, St. Louis, Missouri

SPECIFYING A DESIGN LANGUAGE FOR DIGITAL SYSTEMS. 905
T. Mudge, University of Illinois

A HEURISTIC GUIDED ALGORITHM FOR OPTIMAL BACKBOARD ORDERING 916
A. Sangiovanni-Vincentelli, University of California, Berkeley and M. Santomauro, Politecnico di Milano, Italy

MULTIPLE STACK SEQUENTIAL DECODING ALGORITHM

PIERRE R. CHEVILLAT AND DANIEL J. COSTELLO, JR.
Department of Electrical Engineering, Illinois Institute of Technology
Chicago, Illinois 60616

SUMMARY

Sequential decoding (SD) of convolutional codes is at present the best way of achieving very low error rates over a noisy communication channel. Maximum likelihood decoding (MLD) is less attractive for this purpose because long codes are required and complexity and decoding effort increase exponentially with code constraint length.

The decoding effort of SD, on the other hand, is a random variable with Pareto distribution and is insensitive to constraint length. Typically its average is much lower than the number of computations for MLD. However, a significant number of data frames cannot be decoded in the allocated time, giving rise to erasures due to buffer overflow.

It has been shown that the erasure rate can be reduced by judicious code selection[1] and proper modification of the SD algorithm.[2] However, unless erasures are far more acceptable than errors, an erasurefree ("complete") decoding method like MLD is necessary.

This paper presents a new decoding algorithm which combines the desirable aspects of both MLD and SD:

- the computational effort is a random variable with an average lower than MLD

- decoding is erasurefree, like MLD.

The new algorithm utilizes a multiple stack arrangement in a sequential fashion which is guaranteed to shorten prolonged searches. The analysis of the multiple stack algorithm (MSA) shows that under certain conditions the number of computations has a fixed upper limit. The elimination of erasures is at the cost of a small increase in error probability compared to SD. The average computational effort is still substantially lower than for MLD and the reduced variance significantly lowers decoder idling time.

The performance of the MSA is compared to MLD and ordinary SD and the results of computer simulations are reported.

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation, under Grant ENG 74-181175, at the Illinois Institute of Technology, Chicago, Illinois 60616.

REFERENCES

1. Chevillat, P. R. and D. J. Costello, "Distance and Computation in Sequential Decoding", IEEE Trans. on Communication Tech., to appear.

2. Forney, G. D. and E. K. Bower, "A High-Speed Sequential Decoder: Prototype Design and Test", IEEE Trans. on Communication Tech., COM-19 (Part II), pp 821-835, October 1971.

NOISE MEASUREMENTS AND GOOD CODES FOR MULTI-PATH TREE ENCODED SPEECH

C.-W. LAW
J.B. ANDERSON
R.B. CIRJANIC
Dept. of Elec. Eng. & Communications Research Lab.
McMaster University,
Hamilton, Ontario.

I. INTRODUCTION

Recent work has applied to speech coding the information theoretic idea of tree source coding. Speech digitizers of the waveform-following type, such as PCM, adaptive delta-modulation, and differential PCM, all have inherent in them tree source codes, but these schemes develop only a single word of the code. At the heart of coding theory is the consideration of the entire code. Efficient search algorithms [1]-[3] have been designed for tree source codes. Use of these "multiple-path" approaches on DPCM-like codes (which are close to optimal in a theoretical sense) has yielded a 4-8 dB improvement in signal-to-noise ratio, and certain other advantages, at a modest increase in encoder complexity, and no increase in decoder complexity. Performance at 16 k bits/sec is about 16-20 dB SNR.

In this paper, we discuss first the measurements of encoder noise, as the intensity of searching is increased, and as the power of the input signal becomes more mismatched with the code. We discuss also a computer search for effective codes. Codes with near-optimal performance in a theoretical sense can be obtained using a transversal filter with multiplier coefficients c_i. These c_i's were tested against actual speech by means of a simulation program, which directed a hill-climbing search for better c_i's.

II. THE MULTIPATH ENCODER

For a differential PCM (DPCM) encoder, the predictor is a recursive filter with transfer function

$$H(z) = \frac{1}{1-A(z)} \tag{1}$$

with $A(z) = a_1 z^{-1} + a_2 z^{-2} + \dots + a_k z^{-k}$,

where $\{a_i\}$ are determined from autocorrelation of speech [1].

Fig. 1 illustrates a general multipath encoder. The encoder consists of four important subsystems. They are the distortion calculation unit, predictor (transversal filter), storage, and search algorithm unit. The output of the transversal filter is proved [1] to be identical to the output of the above recursive filter. Hence, the estimated value of signal can be expressed as

$$\hat{x}_t = \sum_{i=1}^{t} c_i q_{t-i} + q_t \tag{2}$$

So long as

$$c_i = \sum_{j=1}^{t} a_j c_{i-j};\ c_o=1,\ c_i=0 \quad \text{for } i < o$$

q_t is a quantization of the estimation error e_t at time t. The $\{c_i\}$ are in fact the impulse response of $\frac{1}{1-A}$. Since the decoded output is the predicted signal $\hat{x}_t$, the decoder is simply a replica of the transversal filter. For details, refer to [1]; in particular, theory suggests a numerator for (1).

The above multipath encoding of speech is based on the so-called M-algorithm (see [1]). Briefly, the algorithm persues a fixed number M of paths at each level throughout the code tree. At each level, all bM branches are extended out of the M saved paths and only the best M of these is saved for the next level, where b is the number of branches at a node of the tree. To ensure finite storage, only λ path digits (the "memory length") are retained. The algorithm amounts to a highly truncated viterbi search.

III. NOISE MEASUREMENT

By neglecting the channel noise of the system, noise analysis can be performed at the encoder. Prior to processing, the analogue speech is sampled at 8 kHz. After passing through an A/D converter, the samples are stored on magnetic tape. The multipath tree encoding system with rate R bits/sample is simulated by computer, and each encoding compared with the original sample.

In measuring the noise, the two most significant effects have been considered. They are the intensity of searching, i.e. the value of M, and the degree of match between the code and the input signal amplitude, the "step size". Encoder noise is analyzed two ways, by an average of many short term spectra (using an FFT), and by autocorrelation using a very large (2 sec.) window.

Fig. 2 shows noise spectra for an encoder which keeps 1,2,4 and 8 code word "paths" in contention at all times. The spectra show that as M increases, the noise power will decrease and tend to flatten. The results verify a common belief of coding theorists, that encoder noise whitens as good codes are more carefully used. The autocorrelation of noise for these four degrees of searching is shown in Fig. 3. The plot shows that as the degree of searching increases, the autocorrelation function of the noise tends to an impulse at zero, indicating uncorrelated noise. Neither figure necessarily implies independent noise; we have not carried out tests to determine this.

The subjective effect of these changes is less irritating noise and lack of sampling "note" at $f_s/2$. The latter must normally be removed by severe filtering. The greatest noise improvement occurs at the formants, which apparently cause the most strain on an ordinary single-path searched code.

Fig. 4 shows spectra of the noise with different step sizes. The better the match of the code and the input power, the smaller the noise power and the flatter the spectrum. Results for power mismatches also indicate that better noise spectra result when the code is larger in amplitude than it should be, rather than the reverse. When the code underestimates power, the noise spectrum follows the input power spectrum. The autocorrelation function for different degrees of power mismatches are plotted in Fig. 5. Noise tends to be most uncorrelated when code and input levels match. Results are again better for codes overestimating the source power, rather than reverse.

All data are from encoding "speed and efficiency were stressed" using the (1,0.6,-0.4) McDonald code of [1].

IV. COMPUTER SEARCH FOR EFFECTIVE SPEECH CODES

Codes with near-optimal performance can be obtained from theoretical considerations (see [1]) using a transversal filter with multiplier coefficients c_i. However, the theory makes certain assumptions, the most damaging of which are single-path code searching, long constraint length, and absence of quantization noise. But better results can be obtained by testing various c_i against actual speech in a hill-climbing search for the best c_i. Since an objective function cannot easily be defined, we cannot find the optimum by applying standard optimization techniques. Instead, we have used the following scheme:

1. Get the values for $\{c_i\}$ which generate codes with near-optimal theoretical performance, process the speech, get SNR and let it be MAX.

2. Increase c_o by Δc and keeping other c's unchanged, process the same speech; get SNR, compare with MAX. If SNR is greater, replace MAX by it. Repeat for all c_i.

3. Repeat above with negative Δc.

4. Go to 2, until no increment of the present size increases MAX.

5. Decrease the increment. Let $\Delta c = \Delta c/A$ where $A>1$. Go to 2 unless $\Delta c<\varepsilon$, where ε is the smallest increment allowed.

6. Best code with this starting point has been attained.

Although the code obtained may not be globally optimum, it is a local optimum. Several starting points have been tried and by comparing the local optima, we can obtain a "global" one.

This optimization scheme has been used with speech sampled at 16 k samples/sec and an encoder operating at rate 1 bit/sample. Results are in Fig. 6. With $A = 2$, $\Delta c = 0.5$, $\varepsilon = 0.02$, the optimal point is attained after 10 iterations. Fig. 7 shows frequency response of the original filter and final filter. The optimization tries to smooth the bumps within the 4 kHz input signal bandwidth and makes the noise decay more rapidly after 4 kHz. Hopefully, the optimization has removed the effect of questionable theoretical assumptions.

V. CONCLUSION

Results of the measurement of noise for multipath tree encoded speech verify that encoder noise whitens and is less correlated as good codes are more carefully used, i.e. when the searching intensity is greater. Results also show that degradation is less when the code overestimates the input amplitude rather than the reverse. Of course, the noise is least when the code matches the input power. By means of a simulation program coupled with a hill-climbing search for the best c's, better codes can be achieved.

REFERENCES

1. J.B. Anderson and J.B. Bodie, "Tree Encoding of Speech", IEEE Trans. Information Theory, vol. IT-21, July 1975, pp. 379-387.

2. J.B. Anderson, "A Stack Algorithm for Source Coding with a Fidelity Criterion", IEEE Trans. Information Theory, vol. IT-20, March 1974, pp. 211-226.

3. J.B. Anderson and F. Jelinek, "A 2-Cycle Algorithm for Source Coding with a Fidelity Criterion", IEEE Trans. Information Theory, vol. IT-19, Jan. 1973, pp. 77-92.

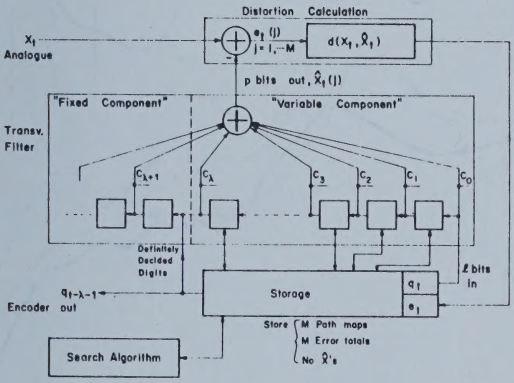

Distortion Calculation
x_t
Analogue
$e_t(j)$
$j = 1, \cdots M$
$d(x_t, \hat{x}_t)$
p bits out, $\hat{x}_t(j)$
"Fixed Component"
"Variable Component"
Transv. Filter
$c_{\lambda+1}$
c_λ
c_3
c_2
c_1
c_0
Definitely Decided Digits
$q_{t-\lambda-1}$
Encoder out
Storage
q_t
e_t
ℓ bits in
Store
M Path maps
M Error totals
No $\hat{x}$'s
Search Algorithm

Fig. 1: Block diagram of multi-path searched tree encoder, using transversal code generator.

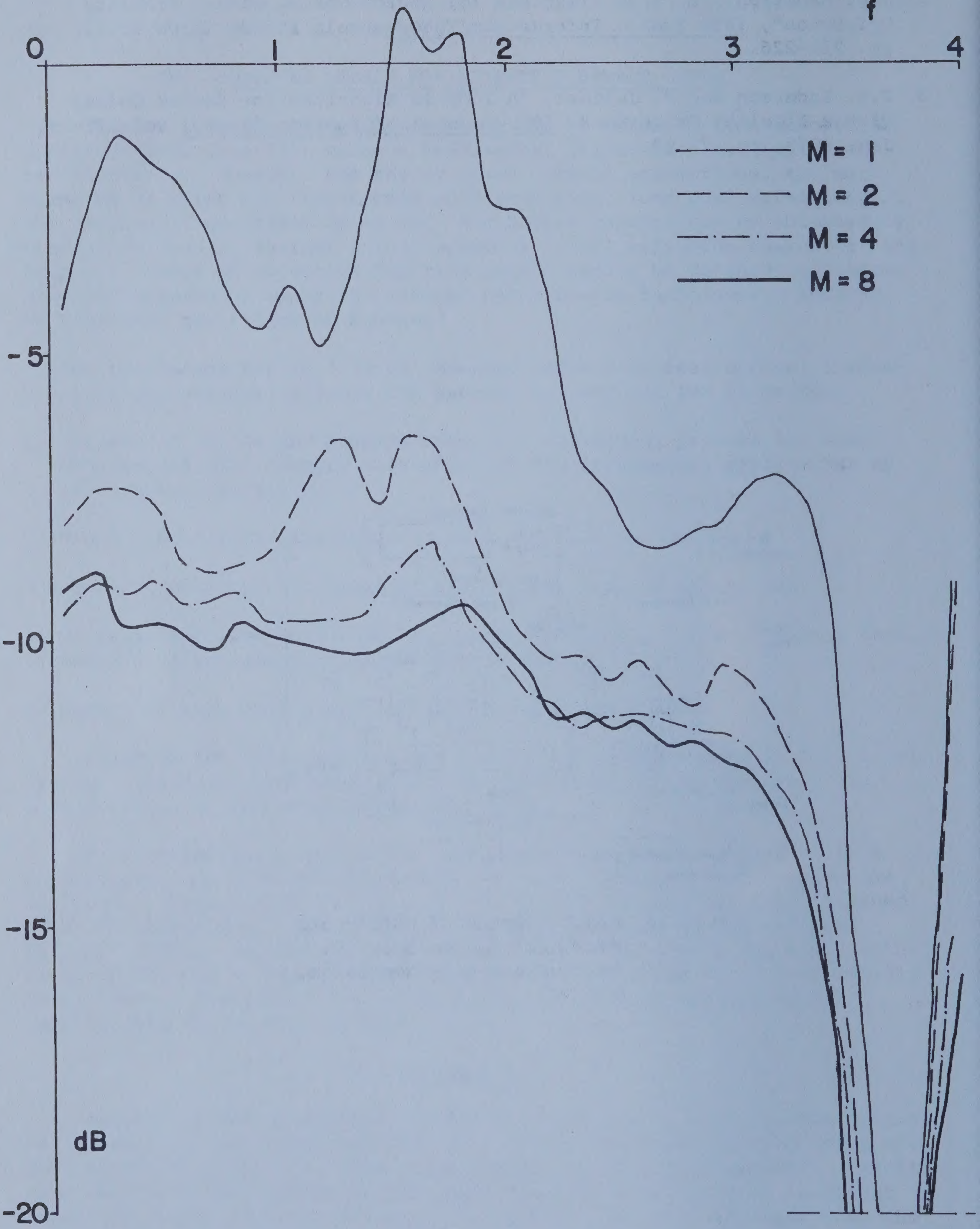


Fig. 2: Power spectrum of noise, dB vs KHz; M=1,2,4,8 with best step size at each M.

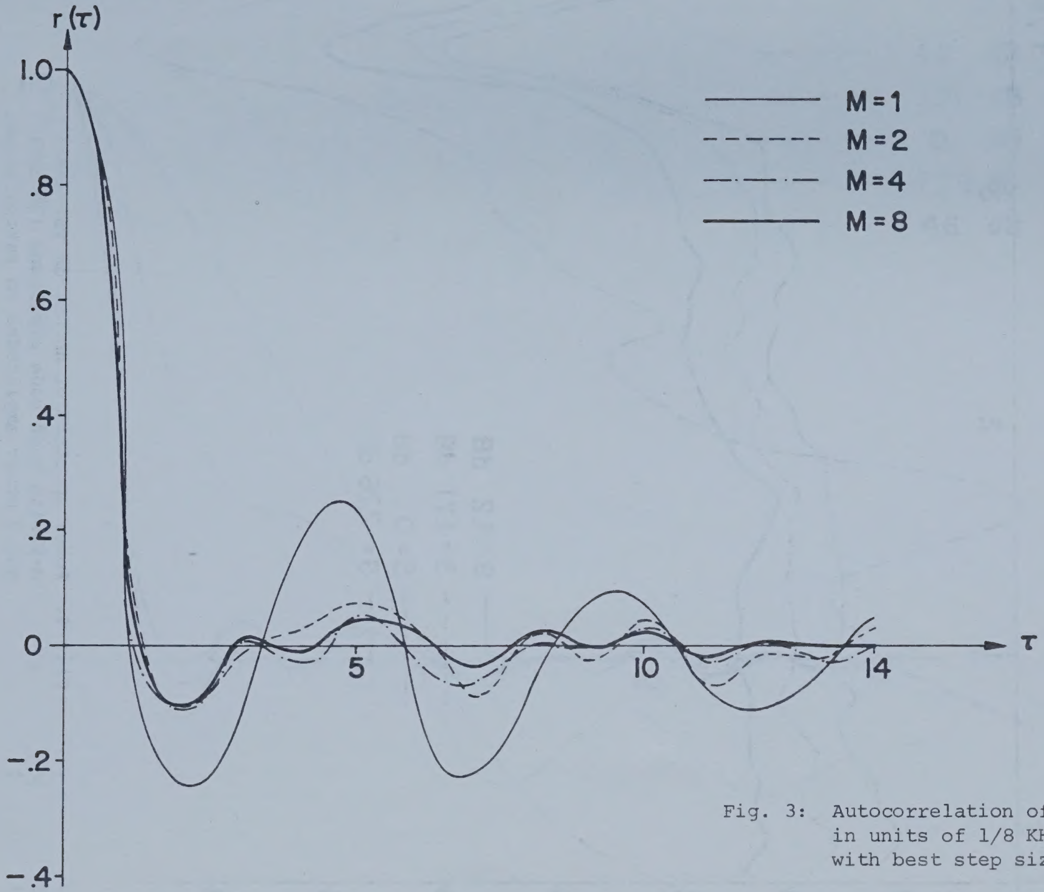


Fig. 3: Autocorrelation of noise, τ in units of 1/8 KHz; M=1,2,4,8 with best step size.

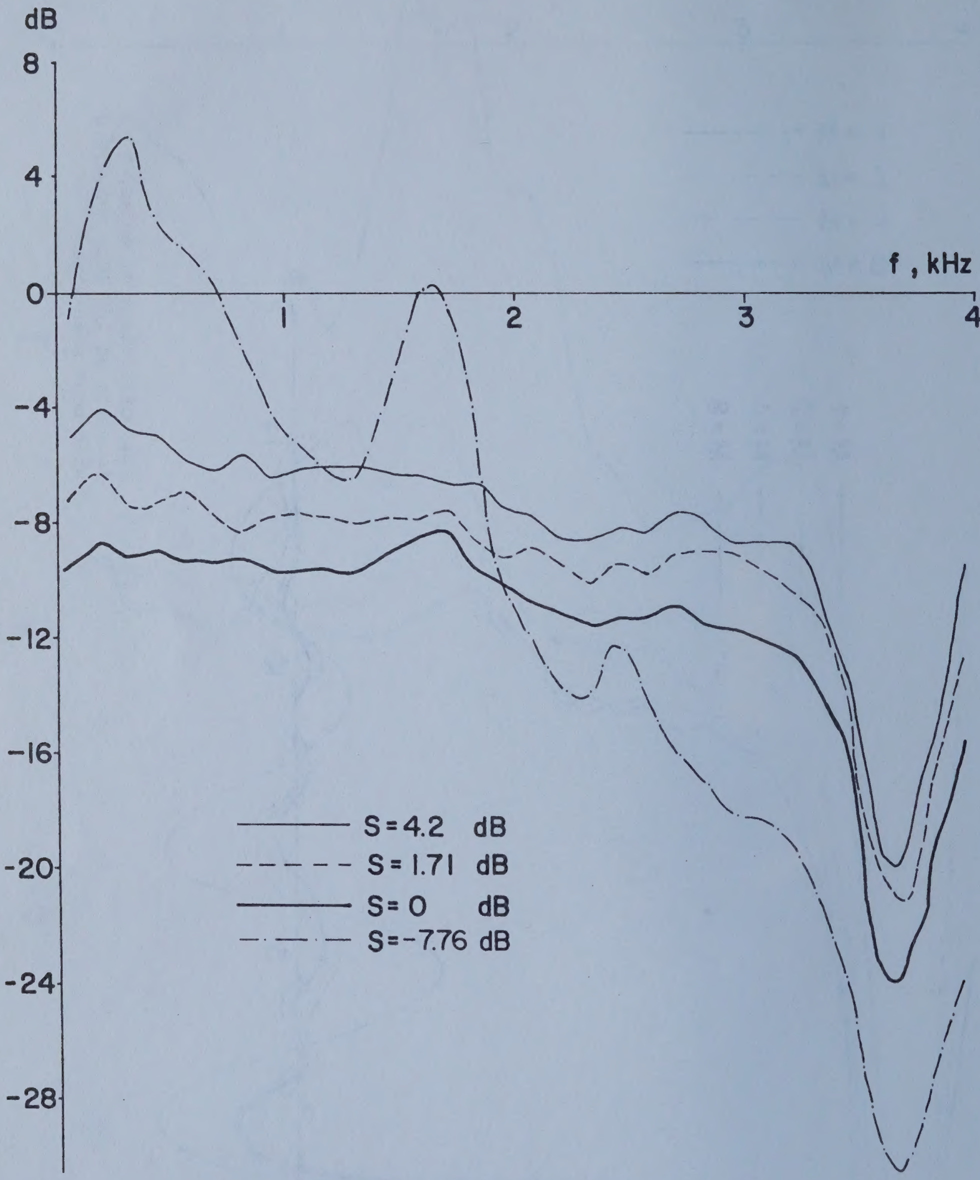


Fig. 4: Power spectrum of noise, dB vs. KHz; M=4 with best code amplitude ("0 dB") and 3 other amplitudes in ratios shown.

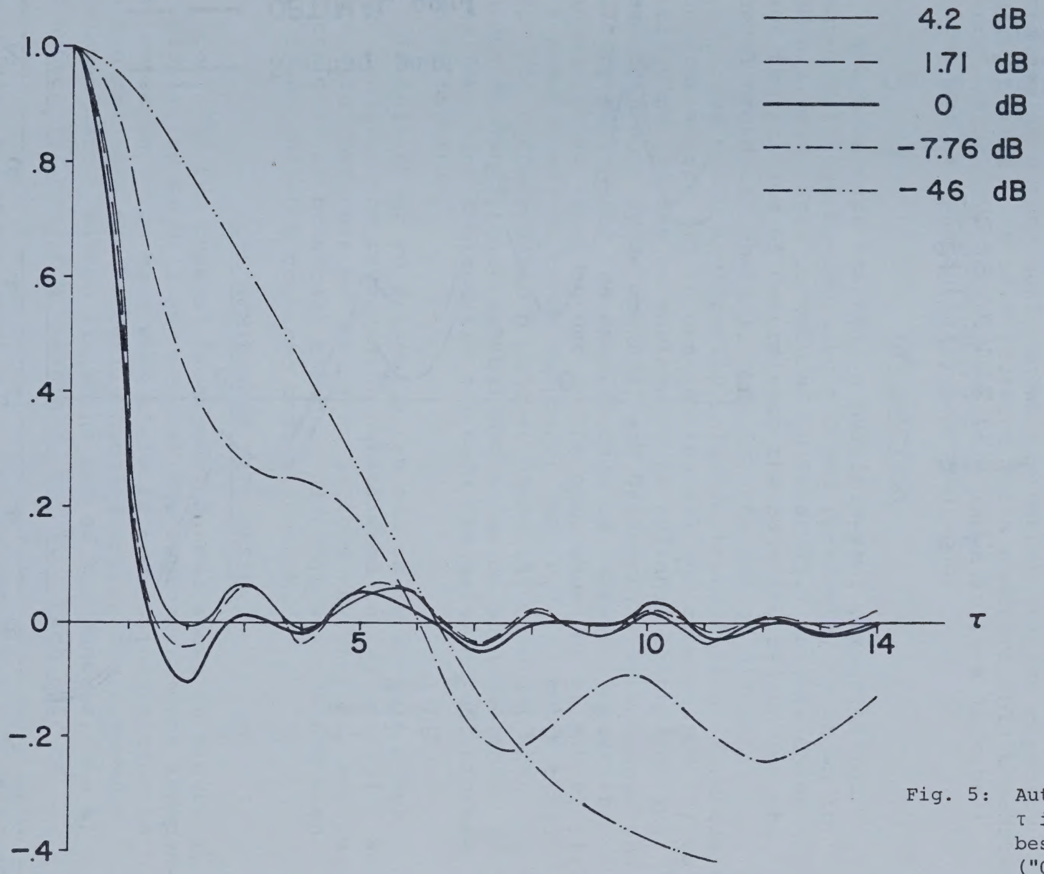


Fig. 5: Autocorrelation of noise, τ in units of 1/8 KHz; best code amplitude ("0 dB") and 4 others.

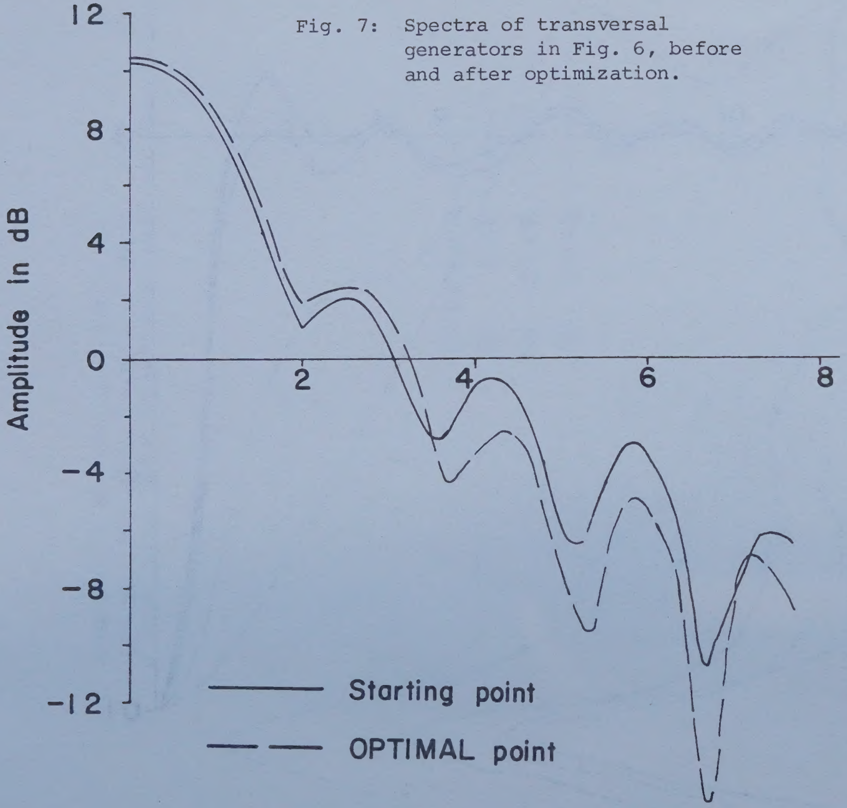


Fig. 7: Spectra of transversal generators in Fig. 6, before and after optimization.

$A = 2, \quad \Delta c = 0.5, \quad \varepsilon = 0.02$

Starting c's	final c's
.645	.676
1.403	1.403
1.504	1.754
1.484	1.796
1.412	1.537
1.262	1.262
1.076	1.076
0.879	0.879
0.677	0.552
0.474	0.255
SNR 14.75	15.50

Fig. 6: Sample run of optimization scheme (run having highest final SNR); f_s=16KHz, R=1, 10 coeff. transversal generator filter.

THE CAPACITY REGION FOR GENERAL BROADCAST CHANNELS*

WILLIAM J. LEIGHTON and HARRY H. TAN
Department of Electrical Engineering, Princeton University
Princeton, New Jersey 08540

ABSTRACT

In this paper, we define a rate region $\mathcal{R}$ and show that arbitrarily reliable transmission of information over a general broadcast channel may be achieved with suitable coding at all rates in $\mathcal{R}$. For the special case of degraded broadcast channels, $\mathcal{R}$ reduces to a previously known expression for the capacity region. Finally, $\mathcal{R}$ is calculated for a nondegradable broadcast channel which shows that previously known achievable rate regions can be substantially improved upon.

INTRODUCTION

The broadcast channel, a multiple-user communication channel with one transmitter and two receivers, as shown in Figure 1, was first introduced by Cover[1]. In [1], Cover posed the problem of determining the capacity region for the general broadcast channel, that is, the set of all rates (R_1, R_2) for which arbitrarily reliable transmission of information from source (1) to destination (1) and from source (2) to destination (2) may be achieved with suitable channel coding. Subsequent work by Bergmans[2] and Gallager[3] established the capacity region for the special case of degraded broadcast channels, that is, when one of the component DMC's, say $P_2(\cdot|\cdot)$ is a cascade of the other component DMC, $P_1(\cdot|\cdot)$ and a third DMC. For nondegradable broadcast channels, Cover[4] and van der Meulen[5] have established a region of achievable rates for which arbitrarily reliable transmission of information is possible.

In this paper we present a new set of achievable rates which contains the rate region established in [4] and [5]. We conjecture that this set of rates defines the capacity region for the general broadcast channel, although we have not been able to establish a converse coding theorem.

THE ACHIEVABLE RATE REGION

Consider the general broadcast channel shown in Figure 1. For simplicity we will assume that the sources produce independent messages. The case when there is common information in the two messages can be easily handled within the framework developed here. Source (1) emits one of M_1 messages, say m, and source (2) one of M_2 messages, say i. Each pair of

---

* This research was supported in part by NSF Grant No. GK-42080 and in part by the Air Force Office of Scientific Research under Grant AFSOR-71-2101.

messages (m,i) is then encoded into a codeword $\underline{x}_{m,i}$ consisting of a sequence of N channel input letters. The codeword $\underline{x}_{m,i}$ is then simultaneously transmitted over DMC's (1) and (2). Decoder (1) receives $\underline{y}$ and attempts to decode m, producing an estimate $\hat{m}$, and decoder (2) receives $\underline{z}$ and attempts to decode i, producing $\hat{i}$. We assume that the two decoders cannot collaborate. Associated with each such block code $C_N = \{\underline{x}_{m,i} : 1 \le m \le M_1, 1 \le i \le M_2\}$ is the probability of decoding error $P_e(C_N) = \max(P_{e,1}(C_N), P_{e,2}(C_N))$ where $P_{e,1}(C_N) = \max_{1 \le m \le M_1} P_r[\hat{m} \ne m | \text{source}(1) \text{ sends } m]$ and $P_{e,2}(C_N) = \max_{1 \le i \le M_2} P_r[\hat{i} \ne i | \text{source}(2) \text{ sends } i]$. A rate pair (R_1, R_2) is said to be achievable if there exists a sequence of codes C_N with $M_1 = \lceil e^{NR_1} \rceil$ and $M_2 = \lceil e^{NR_2} \rceil$ such that $P_e(C_N) \to 0$ as $N \to \infty$. The set of all achievable rate pairs is called the capacity region $\mathcal{C}$ of the broadcast channel.

In this paper we define a rate region $\mathcal{R}$ as follows. Consider a discrete joint ensemble UXYZ with joint probability mass function

$$P_r[U=u, X=x, Y=y, Z=z]$$

$$= Q_1(u)\, Q_2(x|u)\, P_1(y|x)\, P_2(z|x) , \tag{1}$$

where $P_1(\cdot|\cdot)$ and $P_2(\cdot|\cdot)$ are the transition probabilities of DMC(1) and DMC(2) respectively, $Q_1(\cdot)$ an arbitrary probability mass function and $Q_2(\cdot|\cdot)$ an arbitrary conditional probability mass function. Let

$$C_1(Q_1,Q_2) = \{(R_1,R_2) : R_1 < I(U;Y), R_2 < I(X;Z|U)\} \tag{2}$$

and

$$C_2(Q_1,Q_2) = \{(R_1,R_2) : R_1 < I(X;Y|U), R_2 < I(U;Z)\} . \tag{3}$$

Now define

$$\mathcal{R} = \text{convex hull of } \bigcup_{Q_1,Q_2} [C_1(Q_1,Q_2) \cup C_2(Q_1,Q_2)] \tag{4}$$

where the union is over all Q_1 and Q_2 as specified above.

The significance of $\mathcal{R}$ is given in the following theorem.

Theorem 1: Every rate pair in $\mathcal{R}$ is achievable.

Proof: The proof of this theorem is as usual based on a random coding argument. We sketch the gist of the proof below. Details can be found in [6]. We will use a random code ensemble based on the idea of satellite coding introduced in

[2] and [3]. Fix a discrete probability mass function $Q_1(\cdot)$ and a conditional probability mass function $Q_2(\cdot|\cdot)$ as in the joint ensemble (1). An ensemble of codes C_N of blocklength N will be chosen in the following manner. First choose M_2 "cluster centers" $\underline{u}_i$, $1 \le i \le M_2$ with distribution $\prod_{n=1}^{N} Q_1(u_{i,n})$. Then choose M_1M_2 codewords $\underline{x}_{m,i}$, $1 \le m \le M_1$, $1 \le i \le M_2$ by passing each randomly chosen cluster center $\underline{u}_i$ M_1 times through a channel with transition probability $\prod_{n=1}^{N} Q_2(x_{m,i,n}|u_{i,n})$.

The decoding rules used by the decoders will be based upon joint typicality. Let

$$P_3(z|u) = \sum_x P_2(z|x)Q_2(x|u) \quad ,$$

$$P_4(z) = \sum_u P_3(z|u)Q_1(u) \quad ,$$

$$P_5(y|u) = \sum_x P_1(y|x)Q_2(x|u) \quad ,$$

and $$P_6(y) = \sum_u P_5(y|u)Q_1(u) \quad .$$

Define

$$P_{3,N}(\underline{z}\,|\,\underline{u}) = \prod_{i=1}^{N} P_3(z_i|u_i)$$

and $$P_{4,N}(\underline{z}) = \prod_{i=1}^{N} P_4(z_i)$$

for $\underline{z} = (z_1,\ldots,z_N)$ and $\underline{u} = (u_1,\ldots,u_N)$.

Similarly define $P_{1,N}(\underline{y}|\underline{x})$ $P_{5,N}(\underline{y}|\underline{u})$ and $P_{6,N}(\underline{y})$.

Fix an $\varepsilon > 0$. At decoder (2) the received sequence $\underline{z}$ is decoded as i if and only if $\underline{z}$ is ε-typical with $\underline{u}_i$ and not ε-typical with any other cluster center $\underline{u}_k$, $k \ne i$. Here we say $\underline{z}$ is ε-typical with $\underline{u}_j$, where $1 \le j \le M_2$, if $\left| \frac{1}{N} \log \frac{P_{3,N}(\underline{z}|\underline{u}_j)}{P_{3,N}(\underline{z})} - I(U;Z) \right| < \varepsilon$. At decoder (1) the decoding is done in two steps. In the first step, generate a list L of messages k for which the received $\underline{y}$ is ε-typical with $\underline{u}_k$, that is, if $\left| \frac{1}{N} \log \frac{P_{5,N}(\underline{y}|\underline{u}_k)}{P_{6,N}(\underline{y})} - I(U;Y) \right| < \varepsilon$. In the second step, to each m, $1 \le m \le M_1$, assign the $k \in L$ for which

$P_{1,N}(\underline{y}|\underline{x}_{m,k})$ is largest. Call this index $k(m)$. Now decode $\underline{y}$ as m if and only if $\underline{y} \in C_\varepsilon(\underline{x}_{m,k(m)},\underline{u}_{k(m)})$ for a unique m. Here $C_\varepsilon(\underline{x},\underline{u})$ is the set of all received $\underline{y}$ such that

$$\left|\frac{1}{N} \log \frac{P_{1,N}(\underline{y}|\underline{x})}{P_{5,N}(\underline{y}|\underline{u})} - I(X;Y|U)\right| < \varepsilon .$$ In [6] it is shown that the code ensemble average of $P_{e,1}(C_N)$ and $P_{e,2}(C_N)$ can be made smaller than ε for sufficiently large N provided that $(R_1,R_2) \in C_2(Q_1,Q_2)$. This shows the achievability of the rate region $C_2(Q_1,Q_2)$. An identical argument with DMC (1) and DMC (2) interchanged yields the achievability of $C_1(Q_1,Q_2)$. Using time-sharing, we can conclude that $\mathcal{R}$ is achievable. This concludes the proof of Theorem 1.

The rate region $\mathcal{R}$ defined by (4) is not suitable for computation since the size of the U alphabet is not constrained. We now present an alternate representation of the achievable rate region $\mathcal{R}$. Consider an ensemble, UXYZ, satisfying (1). Let I,K,J,L be the sizes of the U,X,Y,Z alphabets respectively. Define

$$C_1(\lambda) = \underset{Q_1Q_2}{\mathrm{MAX}} \; I(U;Z) + \lambda I(X;Y|U) \tag{5}$$

where the size of the U alphabet is constrained to be MIN[K,L]. Similarly, with I = MIN[K,J], let

$$C_2(\lambda) = \underset{Q_1Q_2}{\mathrm{MAX}} \; I(X;Z|U) + \lambda I(U;Y) \tag{6}$$

Letting $C(\lambda) = \mathrm{MAX}[C_1(\lambda),C_2(\lambda)]$ yields the following theorem.

Theorem 2: The outer boundary of $\mathcal{R}$ is described by

$$R_2(R_1) = \underset{\lambda\geq 0}{\mathrm{INF}} \; C(\lambda) - \lambda R_1 \tag{7}$$

The proof is similar to that of Lemma 1 of [3] and will be omitted here. The significance of this theorem is not only the definition of $\mathcal{R}$ as the solution of a nonlinear programming problem but also the considerable reduction in computational effort due to the restrictions on the size of the U alphabet.

A typical $R_2(R_1)$ curve is shown in Figure 2. By definition $R_2(R_1)$ is the lower envelope of the series of straight lines $R_2 = C(\lambda) - \lambda R_1$ and is therefore a closed convex curve in the $R_1 \times R_2$ plane. Note that by defining $C(\lambda) = \mathrm{MAX}[C_1(\lambda),C_2(\lambda)]$, $R_2(R_1)$ automatically takes into account any possibilities of time sharing between rate pairs defined by (2) and those defined by (3). If the channels form a degraded pair, (7) reduces to the expression for the capacity region given in [3]. Thus $\mathcal{R} = \mathcal{C}$ for degraded broadcast channels. We now show that $\mathcal{R}$ contains the achievable rate region for general broadcast channels given in [4] and [5]. In [4] and [5] a discrete joint ensemble $YZXX_1X_2$ with joint probability $P_1(y|x)P_2(z|x)Q(x|x_1x_2)Q_1(x_1)Q_2(x_2)$ for a sample point $yzxx_1x_2$ is considered. The achievable rate region $\mathcal{R}'$ given in [4] and [5] is defined as

$$\mathcal{R}' = \text{convex hull of} \bigcup_{Q,Q_1,Q_2} \{(R_1,R_2): R_1 < I(X_1;Y), R_2 < I(X_2;Z)\} \tag{8}$$

But since X_1 and X_2 are independent,

$I(X_1;Y) \le I(X;Y|X_2)$ and $I(X_2;Z) \le I(X;Z|X_1)$. Thus we see from (2) and (3) that any rate pair in $\mathcal{R}'$ is contained in $\mathcal{R}$.

Before proceeding to some examples, it is instructive to rewrite (3) as

$$C(Q_1,Q_2) = \{(R_1,R_2) : R_1 < I(X;Y) - I(U;Y) \text{ and } R_2 < I(U;Z)\}.$$

This expression for rate pairs gives a more intuitive picture of how satellite coding works. For a fixed R_2, R_1 is the normal capacity of channel (1) minus the amount of channel (1)'s capacity used in transmitting to channel (2). In this sense degraded broadcast channels are undesirable, since if DMC(2) is a degraded version of DMC(1) we always have $I(U;Z) \le I(U;Y)$. Therefore we always subtract off more than R_2 from channel(1). Certain nondegraded channels, on the other hand, can have $I(U;Y)$ very small with $I(U;Z)$ large, in which case we pay very little in terms of R_1 for transmitting to channel(2).

BLACKWELL CHANNEL

An interesting nondegradable broadcast channel is the Blackwell channel which was discussed in [5]. The transition probabilities for this channel are shown in Figure 3. If we set

$$\{Q_2(X|U)\} = \begin{bmatrix} 0 & \frac{1}{2} \\ 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$$

and

$$\{Q_1(U)\} = \begin{bmatrix} p \\ 1-p \end{bmatrix} \tag{9}$$

then (2) becomes

$$C_1(Q_1,Q_2) = \{(R_1,R_2) : R_1 < H(p), R_2 < p\} , \qquad (10)$$

where

$$H(p) = -p\log_2 p - (1-p)\log_2(1-p) .$$

Similarly, let

$$\{Q_2(X|U)\} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix}$$

and $Q_1(\cdot)$ as in (9), then (3) becomes

$$C_2(Q_1,Q_2) = \{(R_1,R_2) : R_1 < p, \ R_2 < H(p)\}. \qquad (11)$$

The convex hull of the union of the two rate regions (10) and (11) is shown in Figure 4. A simple outer bound [5] on the capacity region for this channel is that $R_1 + R_2 \le \log_2 3$. Thus the boundary of the rate region in Figure 4 is seen to have a line segment in common with this outer bound. Moreover the rate region in Figure 4 strictly dominates the rate region given in [5] for this channel.

SUMMARY

In this paper we have presented a new set of achievable rate pairs for general broadcast channels. This achievable rate region reduces immediately to the capacity region for degraded channels and includes previously known achievable rate regions for general broadcast channels. At this time all attempts at proving a non-trivial converse coding theorem have been unsuccessful. However, since $R_2(R_1)$ reduces to the capacity expression for degraded channel and coincides with known outer bounds for some nondegraded capacity regions, it seems reasonable to conjecture that $R_2(R_1)$ defines the capacity region for the general broadcast channel.

REFERENCES

[1] T.M. Cover, "Broadcast channels", IEEE Trans. on Inform. Theory, Vol. IT-18, pp. 2-14, Jan. 1972.

[2] P.P. Bergmans, "Random coding theorems for broadcast channels with degraded components", IEEE Trans. on Inform. Theory, Vol. IT-19, pp. 197-207, March 1973.

[3] R.G. Gallager, "Capacity and coding for degraded broadcast channels", Problems in the Transmission of Information, Vol. 10, No. 3, pp. 3-14, July - Sept. 1974.

[4] T.M. Cover, "An achievable rate region for the broadcast channel", IEEE Trans. on Inform. Theory, Vol. IT-21, pp. 399-404, July 1975.

[5] E.C. van der Meulen, "Random coding theorems for the general discrete memoryless broadcast channel", IEEE Trans. on Inform. Theory, Vol. IT-21, pp. 180-190, March 1975.

[6] W.J. Leighton and H.H. Tan, "The Capacity Region for General Broadcast Channels", to be submitted to IEEE Trans. on Inform. Theory.

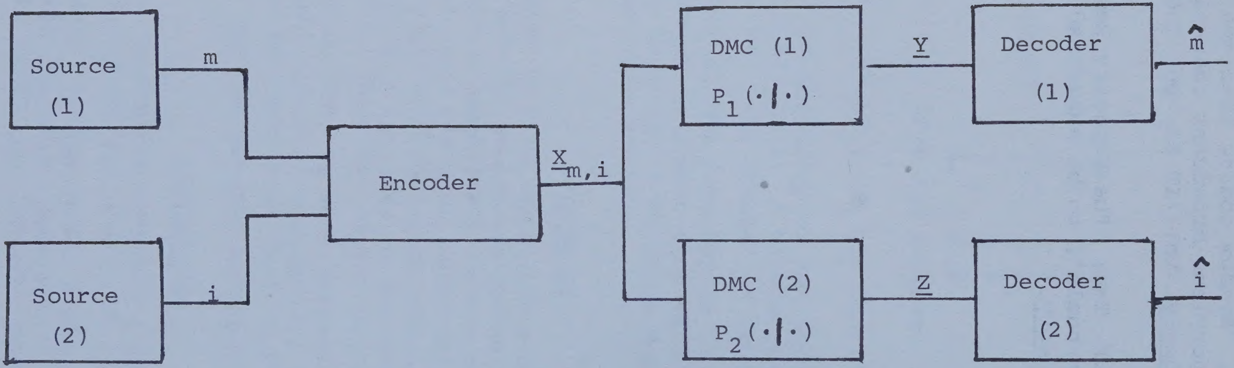

Source
(1)
m
Source
(2)
i
Encoder
$\underline{X}_{m,i}$
DMC (1)
$P_1(\cdot|\cdot)$
DMC (2)
$P_2(\cdot|\cdot)$
$\underline{Y}$
$\underline{Z}$
Decoder
(1)
Decoder
(2)
$\hat{m}$
$\hat{i}$

Figure 1. The General Broadcast Channel

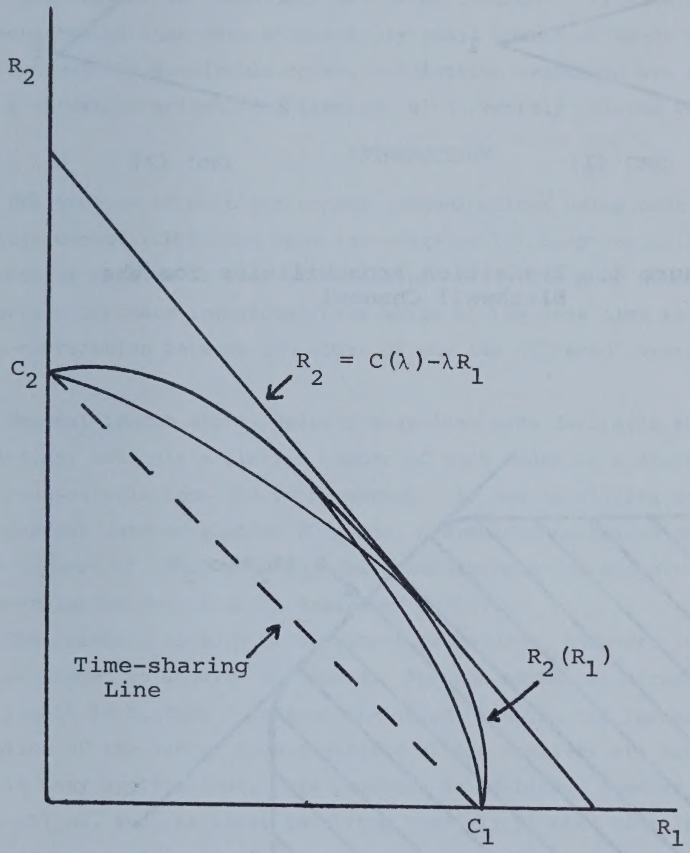

R_2
C_2
$R_2 = C(\lambda)-\lambda R_1$
Time-sharing Line
$R_2(R_1)$
C_1
R_1

Figure 2. Typical $R_2(R_1)$

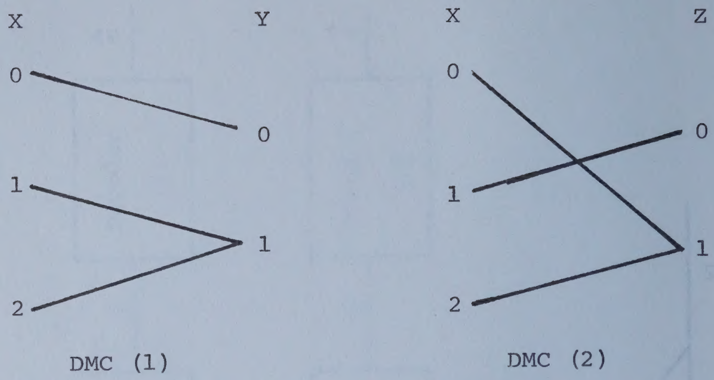

X
Y
0
0
1
1
2
DMC (1)
X
Z
0
0
1
1
2
DMC (2)

Figure 3. Transition Probabilities for the Blackwell Channel

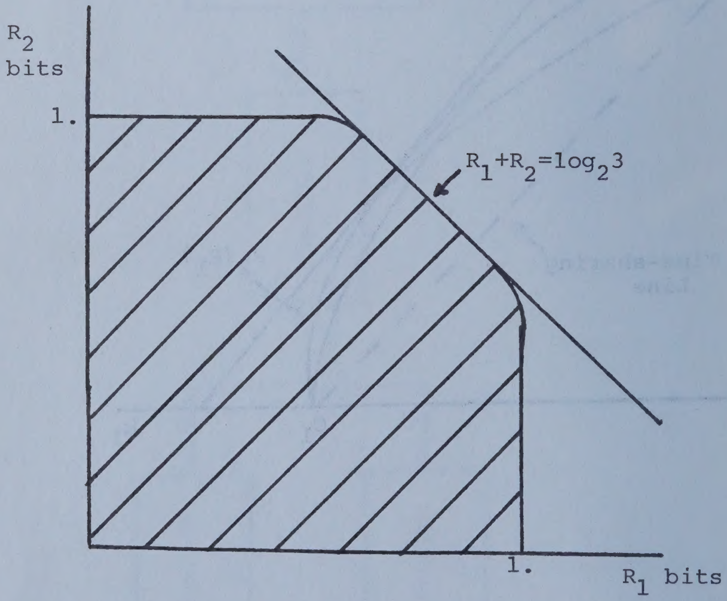


Figure 4. Rate Region for the Blackwell Channel

THE USE OF COMBINATION SEQUENCES IN A MULTIPLE ACCESS ENVIRONMENT

LAURENCE B. MILSTEIN
Electrical and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, New York 12181

ABSTRACT

Combination sequences have long been used successfully for ranging applications. However, very little attempt has been made to determine their performance in a multiple access environment. In this paper, it will be demonstrated that when a relatively small number of users are in need of long but rapidly acquirable codes, combination sequences are a good choice. Such a situation arises, for example, with remotely piloted vehicles.

1. INTRODUCTION

The problem of multiple access communications using code-division multiple access (CDMA) has been investigated for many years ([1]-[3]). The fundamental problem is that of designing codes with good autocorrelation properties for each individual user while at the same time keeping the cross-correlation between the codes of any two different users uniformly low.

Maximal-length shift register sequences have desirable autocorrelation properties, but only a limited number of such codes of a given length with low cross-correlations can be generated. If one is willing to accept a less perfect autocorrelation function, a specific technique of generating large numbers of codes with bounded cross-correlation and out-of-phase autocorrelation functions is described in [3].

The drawback to both of the above procedures, however, is the length of time needed to acquire the codes. For the above two techniques, if the code length is N, then N cross-correlations between the incoming signal and a replica of the users' code generated at the receiver are required.

In many applications, this presents no problem. However, in other applications, such as those involving remotely piloted vehicles (RPV's), speed in acquisition is of the essence. Therefore, the purpose of this paper is to describe such a technique.

The procedure itself has been used in the past to develop quickly acquirable ranging codes ([4], [5]). It consits of generating long codes by combining short sequences on a chip-by-chip basis using a logical combining law. Having been successfully used for ranging applications, it is known that such a combining procedure can lead to long codes with appropriate autocorrelation properties. It remains to investigate whether or not many such sequences can be generated which have desirable cross-correlation properties. This was considered briefly in [2], but no other work (to the best of the author's knowledge) appears to exist.

2. DESCRIPTION OF MODEL

The model to be used is taken from Titsworth ([5]). Given m sequences of lengths n_i, $i = 1, 2, \ldots, m$ where all the n_i are relatively prime, a composite sequence is formed of length $N = \prod_{i=1}^{m} n_i$. Each chip of the composite sequence is a logical combination of one chip from each of the component sequences. For example, suppose the component sequences were given by 110 and 00101, and that the logical function was the "union". The composite sequence would then be as follows:

A = first component 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0
B = second component 0 0 1 0 1 0 0 1 0 1 0 0 1 0 1

C = AUB = composite 1 1 1 1 1 0 1 1 0 1 1 0 1 1 1

In other words, starting with sequences of lengths 3 and 5, a sequence of length 3 x 5 = 15 was generated.

To acquire such a sequence, it is necessary to cross-correlate the composite sequence with each of its components to determine at what relative shift the correlation "peaks up". In particular, the i^{th} component requires n_i such correlations, and therefore a sequence of length $\prod_{i=1}^{m} n_i$ can be acquired with only $\sum_{i=1}^{m} n_i$ correlations. This is in contrast to say a PN sequence where if the length of the sequence was N, N such correlations would be required for acquisition.

The problem then can be stated as follows: If say, J of these composite sequences were being received simultaneously, where each of the J users used independent sequences, how large would the cross-correlation be between the components of one composite sequence and any of the other composite sequences?

Up to this point, no mention was made of the properties of these shorter sequences. It is shown in [5] that these sequences should be balanced in the sense of having approximately equal numbers of ONES AND ZEROES. Therefore, in practice, an obvious choice would be to use PN sequences as the components. However, to make the analysis simpler, random binary sequences with $P(1) = P(0) = \frac{1}{2}$ will be used in this paper.

Also, since in reality ONES and MINUS ONES will actually be transmitted as opposed to ZEROES and ONES, the standard mapping shown below is used to go from the Boolean logic to the binary waveforms.

Binary Sequence		Binary Waveform
0	↔	1
1	↔	-1

The choice of what Boolean function to be used can be made by considering the autocorrelation properties. In [5], Titsworth proved that to maximize the difference between the in-phase correlation value of a composite sequence with one of its component sequences and any out-of-phase correlation value, the Boolean function to choose is that of majority logic. Consequently, in what follows, majority logic will be used.

3. ANALYSIS FOR TOTAL CORRELATION

For simplicity, two short sequences will be used to form the larger sequence. Since majority logic with an even number of components forces some arbitrariness in deciding how to handle ties, the rule used in this paper will be as follows: Using the signum function to implement the majority logic, define

$$\operatorname{sgn}(A + B) \triangleq \begin{cases} 1 & A + B \geq 0 \\ -1 & A + B < 0 \end{cases} .$$

Consider any two composite sequences X and Y. Assume each one was generated by combining two shorter sequences. That is, letting Z_i denote the i^{th} chip of Z,

$$X_i = \operatorname{sgn}(A_{1_i} + A_{2_i})$$

and

$$Y_i = \operatorname{sgn}(B_{1_i} + B_{2_i})$$

where $\{A_{j_i}\}$ and $\{B_{j_i}\}$ are independent random binary sequences of length n_j, $j = 1, 2$, with $i = 1, 2, \ldots, N$.

Assume it is desired to acquire X by first correlating it with A_1. Since Y will be received simultaneously, its (unwanted) contribution will be given by

$$R = \frac{1}{N} \sum_{i=1}^{N} A_{1_i} \operatorname{sgn}(B_{1_i} + B_{2_i}).$$

Consider bounding R by using the Tchebycheff inequality. That is,

$$P[\,|R - \eta| \geq \varepsilon] \leq \frac{\sigma^2}{\varepsilon^2} ,$$

where $\eta = E(R)$, σ^2 is the variance of R, ε is any positive number, and $E(\cdot)$ denotes mathematical expectation.

Since the subsequences are random binary sequences with equally likely symbols, clearly $\eta = 0$. The variance of R can be computed as follows:

$$\sigma^2 = \frac{1}{N^2} E \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} A_{1_i} \operatorname{sgn}(B_{1_i} + B_{2_i}) A_{1_j} \operatorname{sgn}(B_{i_j} + B_{2_j}) \right\}$$

$$= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} E(A_{1_i} A_{1_j}) E\{\operatorname{sgn}(B_{1_i} + B_{2_i}) \operatorname{sgn}(B_{1_j} + B_{2_j})\} \quad (1)$$

There are three cases to consider:

(1) $i \neq j \bmod n_1$

$$E(A_{1_i} A_{1_j}) = 0 \quad (2)$$

(2) $i \equiv j$

$$\frac{1}{N^2} \sum_{i=1}^{N} E(A_{1_i}^{\,2}) E[\operatorname{sgn}^2(B_{1_i} + B_{2_i})] = \frac{1}{N} \quad (3)$$

(3) $i = j \bmod n_1$, $i \not\equiv j$

$$\frac{1}{N^2} \sum_{i=1}^{N} \sum_{\substack{i=j \\ \bmod n_1 \\ i \not\equiv j}} E[A_{1_i} A_{1_j}] E[\operatorname{sgn}(B_{1_i} + B_{2_i}) \operatorname{sgn}(B_{1_j} + B_{2_j})]$$

$$= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{\substack{i=j \bmod n_1 \\ i \not\equiv j}} E(A_{1_i}^2) E[\operatorname{sgn}(B_{1_i} + B_{2_i}) \operatorname{sgn}(B_{1_i} + B_{2_j})] \quad (4)$$

since for $i = j \bmod n_1$,

$$A_{1_i} = A_{1_j},\ B_{1_i} = B_{1_j},\ \text{but } B_{2_i} \neq B_{2_j} .$$

The second expected value in the double summation is easily computed as follows:

$$E[\operatorname{sgn}(B_{1_i} + B_{2_i}) \operatorname{sgn}(B_{1_i} + B_{2_j})]$$

$$= P[B_{1_i} + B_{2_i} \geq 0, B_{1_i} + B_{2_j} \geq 0] + P[B_{1_i} + B_{2_i} < 0, B_{1_i} + B_{2_j} < 0]$$

$$- P[B_{1_i} + B_{2_i} \geq 0, B_{1_i} + B_{2_j} < 0] - P[B_{1_i} + B_{2_i} < 0, B_{1_i} + B_{2_j} \geq 0] = \frac{1}{2}$$

Therefore, (4) reduces to

$$\frac{1}{N^2} \sum_{i=1}^{N} \sum_{\substack{i=j \bmod n_1 \\ 1 \not\equiv j}} (1)\left(\frac{1}{2}\right) = \frac{N(n_2-1)}{N^2}\left(\frac{1}{2}\right) = \frac{n_2-1}{2N} \tag{5}$$

Using (2), (3), and (5) in (1) yields $\sigma^2 = \frac{1}{N} + \frac{n_2-1}{2N} \approx \frac{1}{2n_1}$ for $n_1 \gg 1$, $n_2 \gg 1$. Therefore, the Tchebycheff inequality becomes

$$P\,[|R| \geq \varepsilon] \leq \frac{1}{2n_1\varepsilon^2} \tag{6}$$

4. PARTIAL CORRELATION RESULTS

In [6], similar results are presented for partial correlations, where the partial correlation, R_K, is defined by

$$R_K \triangleq \frac{1}{K} \sum_{i=1}^{K} A_{1_i} \quad \mathrm{sgn}\,(B_{1_i} + B_{2_i}) \, . \tag{7}$$

It is shown in [6] that

$$E(R_K) = E(R) \tag{8}$$

and

$$\mathrm{var}(R_K) \leqq \frac{1}{2K} \left(2 + \left[\frac{K}{n_1}\right]\right) \tag{9}$$

where [x] is defined as the largest integer less than x.

In addition, a much tighter bound on the P $\{R_K > \varepsilon\}$ than that obtained using the Tchebycheff inequality is presented in [6] for the special case of $K \leq \min\,(n_1, n_2)$ using the Chernoff bound (see, e.g., [7], [8]). This latter bound is shown to be valid for any logical combining law (not just for majority logic) as long as the component sequences are balanced.

5. COMBINATION SEQUENCES USED WITH RPV's

The environment that exists with remotely piloted vehicles is one in which a relatively small number of users require long codes to guard against the threat of jamming yet require codes that are rapidly acquirable. As a typical situation, assume there are a total of 10 users. It can be shown that the expected value of the in-phase correlation between a composite sequence and one of its own subsequences is $\frac{1}{2}$ when two subsequences are used. It is reasonable to demand that the total power from the cross-correlations (assuming the powers add) be at least 10 db below that of the desired correlation. With 10 users, this says that ε should eqal $\frac{.5}{10} = .05$. From (6), assuming it is decided to use a sequence of total length $N = 10^9$, and assuming that $n_1 = \sqrt{N} \approx 3 \times 10^4$, we have

$$P\,[\,|R| \geq .05] \leq \frac{1}{2(3 \times 10^{4})(25 \times 10^{-4})} = \frac{1}{150} = 6.7 \times 10^{-3}.$$

To put this result in perspective, it should be pointed out that if the sequences described in [3] were used with a length of approximately 10^9 (these latter sequences have to be of length 2^k-1 for some integer k), the bound on the cross-correlation peaks would be several orders of magnitude smaller than .05. Consequently, many more users could be accommodated. On the other hand, the time needed to acquire the codes described in [3] would be orders of magnitude larger than the time needed to acquire the combination sequences.

Also, it should be pointed out that one could be expected to do reasonably better than (6) indicates since the Tchebycheff inequality is in general not a particularly tight bound. To obtain some feeling for this, consider bounding a sum S of independent equally likely binary numbers. Keeping both the probability the same as above and the code length N the same, we have

$$P\,[\,|S| \geq \varepsilon] \leq \frac{\sigma_s^{\,2}}{\varepsilon^2}$$

and

$$\sigma_s^{\,2} = \frac{1}{N}$$

resulting in

$$\varepsilon = \left[\frac{10^{-9}}{6.7 \times 10^{-3}}\right]^{1/2} = .38 \times 10^{-3}$$

By comparison, if one uses the Chernoff bound for the same sum of independent random binary numbers, one obtains (after some algebra)

$$P[S \geq \varepsilon] \leq \frac{1}{2^N} \left[\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{\frac{1-\varepsilon}{2}} + \left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{\frac{1+\varepsilon}{2}}\right]^N$$

$$\left\{ \frac{1}{2} \left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{\frac{\varepsilon}{2}} \left[\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{.5} + \left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{.5}\right] \right\}^N \tag{10}$$

For $\varepsilon \ll 1$, the left-hand side of (10) can be approximated by

$$(1 - \varepsilon)^{N\varepsilon}$$

Using the values $P\,[S > \varepsilon] = 6.7 \times 10^{-3}$ and $N = 10^{+9}$ yields $\varepsilon = .71 \times 10^{-4}$ or an improvement of more than a factor of five.

While one certainly cannot say the same factor of five will apply when one is bounding the sum of correlated random variables, it does appear likely that some improvement will occur.

6. SUMMARY

This paper has explored the possibility of extending a technique which has been used for many years in a ranging environment to a multiple access environment. The main result can be summarized as follows: Given a relatively small number of users, combination sequences are a very attractive choice. As stated in the preceding paragraphs, one example of such a situation is that involving RPV's.

7. REFERENCES

[1] A. R. Cohen, J. A. Heller, and A. J. Viterbi, "A New Coding Technique for Asynchronous Multiple Access Communication", IEEE Trans. Comm. Tech., Oct. 1971, pp. 849-855.

[2] D. J. Braverman, "Code Division Multiple Access Communication", Tech. Internal Correspondence, Hughes Aircraft Co., April 1964, Ref. 2230.5/179.

[3] R. Gold, "Optimal Binary Sequences for Spread Spectrum Multiplexing", IEEE Trans. Info. Th., Oct. 1967, pp. 619-621.

[4] L. Baumert, M. Easterling, S. W. Golomb, and A. J. Viterbi, "Coding Theory and Its Applications to Communication Systems", JPL Tech. Report No. 32-67, March 1961.

[5] R. C. Titsworth, "Optimal Ranging Codes", IEEE Trans. Space Electronics and Telemetry, March 1964, pp. 19-39.

[6] L. B. Milstein, "Some Statistical Properties of Combination Sequences and Their Use in a Multiple Access Environment", submitted to IEEE Trans. Info. Th.

[7] R. G. Gallager, Information Theory and Reliable Communication, Wiley, 1968.

[8] J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering, Wiley, 1965.

* This research has been partially supported by the Air Force Office of Scientific Research (AFSC) under contract No. AFOSR-73-2486

BIDIRECTIONAL CONVOLUTIONAL CODES

AHMAD ELAHI-TALEGHANI AND DANIEL J. COSTELLO, JR.
Department of Electrical Engineering, Illinois Institute of Technology
Chicago, Illinois 60616

SUMMARY

A new class of rate ½ binary convolutional codes is introduced. These codes are similar in structure to the quick-look-in codes of Massey and Costello[1] and have distance properties comparable to the complementary codes of Bahl and Jelinek.[2] The class is specified by choosing the generator polynomials $G^{(1)}(D) = g_o^{(1)} + g_1^{(1)}D + \cdots + g_m^{(1)}D^m$ and $G^{(2)}(D) = g_o^{(2)} + g_1^{(2)}D + \cdots + g_m^{(2)}D^m$ such that

$$g_o^{(1)} = g_o^{(2)} = g_m^{(1)} = g_m^{(2)} = 1 \tag{1A}$$

$$G^{(1)}(D) + G^{(2)}(D) = D + D^{m-1}, \tag{1B}$$

where m is the memory order of the code and addition is modulo-2. The quick-look-in codes are specified by choosing

$$G^{(1)}(D) + G^{(2)}(D) = D, \tag{2}$$

whereas for the complementary codes

$$G^{(1)}(D) + G^{(2)}(D) = D + D^2 + \cdots + D^{m-1}. \tag{3}$$

Because of the symmetry of eqns. (1A) and (1B), we refer to the new codes as bidirectional codes.

Bidirectional codes have been constructed using a computer construction algorithm out to m = 23. Codes with free distance 24 were found at m = 23. This is the same free distance achievable by complementary codes with m = 23. The quick-look-in codes, on the other hand, fall off in free distance after about m = 12, and can achieve a free distance of only 19 at m = 23.

These results illustrate that in order for a class of codes to have good distance properties it must be capable of having large minimum distance in both the forward and reverse directions. Both the complementary and bidirectional codes have this property, the quick-look-in codes do not. The forward minimum distance of quick-look-in codes meets the Gilbert bound, but the reverse minimum distance is always only 3. This results in a low free distance for these codes.

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation under Grant ENG 74-181175, at the Illinois Institute of Technology, Chicago.

REFERENCES

1. Massey, J. L. and D. J. Costello, "Nonsystematic Convolutional Codes for Sequential Decoding in Space Applications", IEEE Trans. Comm. Tech., COM-19, No. 5 (Part II), pp 806-813, October 1971.
2. Bahl, L. R. and F. Jelinek, "Rate ½ Convolutional Codes with Complementary Generators", IEEE Trans. Inform. Th., IT-17, No. 6, pp 718-727, November 1971.

ROUTH APPROXIMATION METHOD AND SINGULAR PERTURBATIONS

M. F. HUTTON
The Singer Company, Kearfott Division
Little Falls, New Jersey

ABSTRACT

The Routh approximation method (RAM) is a general algorithm for reducing the order of linear, time-invariant systems that utilizes the well-known Routh table of stability theory. This algorithm defines, for any state space representation, a transformation of the state vector that partitions the differential equations into "slow" and "fast" variables. Consequently, the RAM can be formulated as a special singular perturbation method that includes a systematic identification of the small parameter which when set to zero yields the reduced order model. The RAM also possess other valuable approximation properties and is illustrated by a 4th order example.

INTRODUCTION

The Routh approximation method (RAM) is a technique for reducing the order of a linear, time-invariant system that is based on truncating the Routh table normally used to determine stability. Singular perturbation theory is an approach for reducing the order of dynamic systems that is based on approximating certain differential equations by algebraic equations. This paper shows that the RAM belongs to the class of singular perturbation methods although it was not originally developed from this perspective.

A singular perturbation problem for a system of linear, time-invariant differential equations has the vector-matrix form

$$\begin{aligned} \dot{w} &= A_1(\epsilon)w + A_2(\epsilon)z + B_1(\epsilon)u \\ \epsilon\dot{z} &= A_3(\epsilon)w + A_4(\epsilon)z + B_2(\epsilon)u \end{aligned} \tag{1}$$

with observations

$$y = C_1(\epsilon)w + C_2(\epsilon)z$$

In accordance with the well-known theory, the small scalar ϵ partitions the total state vector of dimension n into k "slow" variables w and $(n-k)$ "fast" variables z. The coefficient matrices in (1) may also be a function of ϵ. Setting ϵ to zero reduces the order from n to k, since $(n-k)$ of the differential equations are replaced by algebraic equations. Let $w(t,\epsilon)$ and $z(t,\epsilon)$ denote the solution to (1) for nonzero ϵ

and, for the initial conditions w_0 and z_0 at time $t=0$ and let $\overline{w}(t)$ and $\overline{z}(t)$ denote the solution to the reduced order system obtained from (1) by setting $\epsilon = 0$ and with $\overline{w}(0) = w_0$. One could reasonably expect that

$$\begin{aligned} w(t,\epsilon) &\approx \overline{w}(t) \\ z(t,\epsilon) &\approx \overline{z}(t) + z(\tau) - \overline{z}(0) \end{aligned} \tag{2}$$

where $z(\tau)$ is the solution to the "boundary layer system"

$$\frac{dz}{d\tau} = A_3(0)w_0 + A_4(0)z + B_2(0)u_0 \tag{3}$$

for

$$u(0) = u_0 \quad , \qquad z(0) = z_0$$

If the eigenvalues of the matrix $A_4(0)$ in (3) all have negative real parts, then from the theory of singular perturbations [1] , [2] the approximation (2) is valid and of order ϵ . This is a valuable approximation method once the problem is in the partitioned form shown in (1).

One of the difficulties [2], [3] in applying singular perturbation methods is that the small parameter(s) ϵ , which leads to the partitioned form (1), must be known a priori. In general, an $\underline{n}$th order, linear, time-invariant model of a physical system will have the vector-matrix form

$$\begin{aligned} \dot{x} &= Fx + Gu \\ y &= Px \end{aligned} \tag{4}$$

There are many cases in which one knows that the order of the model in (4) can be reduced but the equations are not in the form of (1) because ϵ is not isolated. To isolate ϵ one must find a transformation of state variables that displays the small parameter ϵ as in (1). In other words, one must find a matrix T such that

$$q = \begin{bmatrix} w \\ -\tilde{z}- \\ z \end{bmatrix} \begin{matrix} \updownarrow k \\ \\ \updownarrow n-k \end{matrix} = Tx \tag{5}$$

with $\qquad \tilde{z} = \epsilon z \qquad (6)$

and where the transformation determines ϵ . It should be noted that T and ϵ are not unique.

ROUTH APPROXIMATION METHOD (RAM)

In this paper we demonstrate that the RAM developed previously [4] - [7] is a singular perturbation method in which the matrix T is directly computed and the small parameter ϵ is systematically identified. The formulation of the RAM in the framework of singular perturbation theory is divided into three basic steps as described below. In order to reduce the complexity, the development of the RAM in this paper will be limited to single input-single output systems that are asymptotically stable. The extention of the RAM to unstable and multiple input-multiple output systems is described in [5] -[7].

Step 1: Transform to Routh Canonical Representation.

Starting from (4) find a state vector transformation (5) that, when substituted into (4), gives a state space representation of the following form:

$$\begin{aligned} \dot{q} &= Aq + Bu \\ y &= Cq \end{aligned} \qquad (7)$$

where

$$\begin{aligned} A &= TFT^{-1} = \Gamma^{-1}\Lambda \\ B &= TG = -\Gamma^{-1}E \\ C &= PT^{-1} = \mathfrak{B} \end{aligned} \qquad (8)$$

and where

$$\Lambda = \mathrm{Diag}[\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_n]$$

$$E = [1, 0, 0, \ldots, 0]'$$

$$\mathfrak{B} = [\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_n]$$

$$\Gamma = \begin{bmatrix} -1 & -1 & & & \bigcirc \\ 1 & 0 & -1 & & \\ & 1 & \ddots & \ddots & \\ \bigcirc & & \ddots & 0 & -1 \\ & & & 1 & 0 \end{bmatrix} \qquad (9)$$

$$\Gamma^{-1} = \begin{bmatrix} -1 & 0 & -1 & 0 & -1 & \bullet & 0 & -1 \\ 0 & 0 & 1 & 0 & 1 & \bullet & 0 & 1 \\ -1 & -1 & -1 & 0 & -1 & \bullet & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & \bullet & 0 & 1 \\ -1 & -1 & -1 & -1 & -1 & \bullet & 0 & -1 \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ -1 & -1 & -1 & -1 & -1 & \cdots & -1 & -1 \end{bmatrix} \quad \text{for } n \text{ odd}$$

The inverse of Γ for n even can be obtained by computing the inverse for $n+1$ using the above formula and then deleting the first row and column. There is a finite recursive algorithm for computing the matrix T tha transforms (4) into the Routh canonical representation given by (7) - (9). A description of this algorithm is given in [5]. For an asymptotically stable system, the n alpha coefficients $\hat{\alpha}_i$ will all be positive and can be computed from the denominator of the transfer function using the Routh table. The n beta coefficients $\hat{\beta}_i$ can be computed from the numerator of the transfer function using a table similar to the Routh table (see [4] - [7]).

Step 2: Partition Routh Canonical Representation.

Suppose the (stable) system matrix F in (4) has k eigenvalues which are small in magnitude relative to the remaining $(n-k)$ eigenvalues. Then the first k alpha coefficients $\hat{\alpha}_1, \ldots, \hat{\alpha}_k$ will be small compared to the remaining $(n-k)$ alpha coefficents. Pick the small parameter ϵ to roughly equal the ratio in magnitudes between the two groups of alpha coefficients, i. e.,

$$\epsilon \approx \hat{\alpha}_1 / \hat{\alpha}_{k+1}$$

Next define $(n-k)$ constants d_i according to

$$d_i = \epsilon \hat{\alpha}_{k+i} \quad , \quad i = 1, 2, \ldots, n-k \tag{10}$$

The introduction of the weighting constants d_i permits each of the small terms $1/\hat{\alpha}_{k+i}$ to be expressed as a linear function of a single small parameter ϵ. Since all the alpha coefficients are positive, ϵ and d_i will also be positive. The matrices defined in (9) are partitioned according to the slow and fast variables as shown in (5). This gives

$$\Gamma = \begin{array}{cc} \begin{array}{cc} \;k\; & \;n-k\; \end{array} & \\ \left[\begin{array}{c:c} \Gamma_1 & \Gamma_2 \\ \hdashline \Gamma_3 & \Gamma_4 \end{array}\right] & \begin{array}{l} k \\ n-k \end{array} \end{array} \qquad \Lambda = \begin{array}{cc} \begin{array}{cc} \;k\; & \;n-k\; \end{array} & \\ \left[\begin{array}{c:c} \Lambda_1 & 0 \\ \hdashline 0 & \Lambda_2 \end{array}\right] & \begin{array}{l} k \\ n-k \end{array} \end{array} \tag{11}$$

$$\Gamma^{-1} = \begin{array}{cc} \left[\begin{array}{c:c} \Omega_2 & \Omega_2 \\ \hdashline \Omega_3 & \Omega_4 \end{array}\right] & \begin{array}{l} k \\ n-k \end{array} \end{array} \qquad \mathfrak{B} = \begin{array}{c} \left[\begin{array}{c:c} \mathfrak{B}_1 & \mathfrak{B}_2 \end{array}\right] \\ \begin{array}{cc} \;k\; & \;n-k\; \end{array} \end{array} \qquad E = \begin{array}{cc} \left[\begin{array}{c} E_1 \\ \hdashline 0 \end{array}\right] & \begin{array}{l} k \\ n-k \end{array} \end{array}$$

Combining (5), (7), (8), and (11) gives that

$$\begin{aligned} \dot{w} &= \Omega_1\Lambda_1 w + \Omega_2\Lambda_2\tilde{z} + \Omega_1 E_1 u \\ \dot{\tilde{z}} &= \Omega_3\Lambda_1 w + \Omega_4\Lambda_2\tilde{z} + \Omega_3 E_1 u \\ y &= \mathfrak{B}_1 w + \mathfrak{B}_2\tilde{z} \end{aligned} \tag{12}$$

From (10) we have that

$$\tilde{z} = \epsilon z = \Lambda_2^{-1} D z \tag{13}$$

with

$$D = \mathrm{Diag}[d_1, d_2, \ldots, d_{n-k}]$$

Substituting (13) into (12) yields a singular perturbation problem of the form shown in (1). Thus, the coefficient matrices in (1) for the RAM are

$$\begin{aligned} A_1 &= \Omega_1\Lambda_1 & B_1 &= \Omega_1 E_1 \\ A_2 &= \Omega_2 D & B_2 &= \Omega_3 E_1 \\ A_3 &= \Omega_3\Lambda_1 & C_1 &= \mathfrak{B}_1 \\ A_4 &= \Omega_4 D & C_2 &= \epsilon\mathfrak{B}_2 \end{aligned} \tag{14}$$

where

$$\begin{aligned} \Omega_1 &= \Gamma_1^{-1} + \Gamma_1^{-1}\Gamma_2\Omega_4\Gamma_3\Gamma_1^{-1} & \Omega_2 &= -\Gamma_1^{-1}\Gamma_2\Omega_4 \\ \Omega_3 &= -\Omega_4\Gamma_3\Gamma_1^{-1} & \Omega_4 &= (\Gamma_4 - \Gamma_3\Gamma_1^{-1}\Gamma_2)^{-1} \end{aligned} \tag{15}$$

The matrix inverses in (15) always exist. In (14) only the matrix C_2 is a function of ϵ.

Step 3: Obtain Reduced Order Model.

Setting ϵ to zero in the differential equations defined by (1) and (14) gives with the aid of (15) the $\underline{k}$th order Routh approximant

$$\begin{aligned}\dot{\bar{w}} &= \Gamma_1^{-1}\Lambda_1\bar{w} + \Gamma_1^{-1}E_1\bar{u}\\ \bar{z} &= D^{-1}\Gamma_3\Gamma_1^{-1}\Lambda_1\bar{w} + D^{-1}\Gamma_3\Gamma_1^{-1}E_1\bar{u}\\ \bar{y} &= \mathfrak{B}_1\bar{w}\end{aligned} \tag{16}$$

Observe that the $\underline{k}$th Routh approximant is a function only of the first k alpha and beta coefficients.

It is not difficult to show that the matrices Ω_4 and D have the same structure as Γ^{-1} and Λ, respectively, only the dimensions are different. Since the diagonal elements of D are all positive, it follows from the Routh stability test that the eigenvalues of $A_4 = \Omega_4 D$ all have negative real parts. Therefore, the approximation in (2) holds for the RAM.

ILLUSTRATIVE EXAMPLE

To illustrate the RAM given by (7) - (16) consider the following 4th order example where the matrices in the state space representation (4) are

$$F = \begin{bmatrix} 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -120 & -180 & -102 & -18 \end{bmatrix} \qquad G = \begin{bmatrix} 0\\ 0\\ 0\\ 1 \end{bmatrix} \qquad P = \begin{bmatrix} 1200\\ 900\\ 248\\ 14 \end{bmatrix}' \tag{17}$$

The input-output transfer function for this example is

$$H(s) = P(s\mathcal{I} - F)^{-1}G = \frac{14s^3 + 248s^2 + 900s + 1200}{s^4 + 18s^3 + 102s^2 + 180s + 120} \tag{18}$$

The first step in applying the RAM is to transform (17) to the Routh canonical representation. The required transformation matrix and its inverse for this example are

$$T = \begin{bmatrix} 180 & 0 & 18 & 0 \\ 0 & 90 & 0 & 1 \\ 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T^{-1} = \begin{bmatrix} 1/180 & 0 & -1/160 & 0 \\ 0 & 1/90 & 0 & -1/90 \\ 0 & 0 & 1/16 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{19}$$

Performing the matrix multiplications in (8) gives

$$A = \begin{bmatrix} 0 & 2 & 0 & 16 \\ -2/3 & -2 & 0 & -16 \\ 0 & 0 & 0 & 16 \\ -2/3 & -2 & -45/8 & -16 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} 20/3 \\ 10 \\ 8 \\ 4 \end{bmatrix}' \tag{20}$$

Comparing (20) with (8) and (9) gives that the alpha and beta coefficients are

$$\begin{array}{ll} \hat{\alpha}_1 = 2/3 & \hat{\beta}_1 = 20/3 \\ \hat{\alpha}_2 = 2 & \hat{\beta}_2 = 10 \\ \hat{\alpha}_3 = 45/8 & \hat{\beta}_3 = 8 \\ \hat{\alpha}_4 = 16 & \hat{\beta}_4 = 4 \end{array} \tag{21}$$

It should be pointed out that the alpha coefficients can be determined by computing the Routh table as illustrated below for the denominator in (18)

$$\begin{array}{l|lll} & 120 & 102 & 1 \\ & 180 & 18 & \\ \hline \hat{\alpha}_1 = 120/180 = 2/3 & 102 - 18\hat{\alpha}_1 = 90 & 1 - 0 \cdot \alpha_1 = 1 & \\ \hat{\alpha}_2 = 180/90 = 2 & 18 - 1\hat{\alpha}_2 = 16 & & \\ \hat{\alpha}_3 = 90/16 = 45/8 & 1 - 0 \cdot \hat{\alpha}_3 = 1 & & \\ \hat{\alpha}_4 = 16/1 = 16 & & & \end{array} \tag{22}$$

The alpha coefficients are the ratios of the elements in the first column (to the right of the vectical line). Hence, from the Routh stability test, the system is asymptotically stable if and only if all the alpha coefficients are positive. Since the F matrix in this example is in companion form, the nonzero elements in the transformation matrix T can also be obtained directly from the elements in the Routh table that are to the right of the vectical line. The matrix T in (19) is an upper triangular matrix in which the main diagonal is the first column of the Routh table (22) starting with the second row. The next or second upper diagonal is zero. The third upper diagonal is the second column of the Routh table starting with the second row. The forth or last upper diagonal of T is zero. The inverse of T as illustrated in (19) also has the same structure of an upper triangular matrix in which the upper diagonals alternate between nonzero and zero. The nonzero elements of T^{-1} can be computed from the nonzero elements of T by a simple recursive algorithm.

The next step in the RAM is to partition the matrices A , B , C in (19) in accordance with the order of the approximation k . This partitioning reduces the problem to a singular perturbation problem as illustrated below for the case k=2 .

$$\begin{gathered}
\begin{bmatrix}\dot{w}_1\\ \dot{w}_2\end{bmatrix} = \begin{bmatrix}0 & 2\\ -2/3 & -2\end{bmatrix}\begin{bmatrix}w_1\\ w_2\end{bmatrix} + \begin{bmatrix}0 & 2\\ 0 & -2\end{bmatrix}\begin{bmatrix}z_1\\ z_2\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}u \\
\epsilon\begin{bmatrix}\dot{z}_1\\ \dot{z}_2\end{bmatrix} = \begin{bmatrix}0 & 0\\ -2/3 & -2\end{bmatrix}\begin{bmatrix}w_1\\ w_2\end{bmatrix} + \begin{bmatrix}0 & 2\\ -45/64 & -2\end{bmatrix}\begin{bmatrix}z_1\\ z_2\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}u \\
y = [20/3,\ 10]\begin{bmatrix}w_1\\ w_2\end{bmatrix} + \epsilon[8\ ,\ 4]\begin{bmatrix}z_1\\ z_2\end{bmatrix}
\end{gathered} \tag{23}$$

where $\epsilon = 1/8 \approx \hat{\alpha}_1/\hat{\alpha}_3$

$d_1 = \epsilon\hat{\alpha}_3 = 45/64$ $\qquad d_2 = \epsilon\hat{\alpha}_4 = 2$

Setting $\epsilon = 0$ in (23) gives the 2nd order Routh approximant

$$\begin{gathered}
\begin{bmatrix}\dot{\bar{w}}_1\\ \dot{\bar{w}}_2\end{bmatrix} = \begin{bmatrix}0 & 2\\ -2/3 & -2\end{bmatrix}\begin{bmatrix}\bar{w}_1\\ \bar{w}_2\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\bar{u} \\
\begin{bmatrix}\bar{z}_1\\ \bar{z}_2\end{bmatrix} = \begin{bmatrix}-128/135 & -128/45\\ 0 & 0\end{bmatrix}\begin{bmatrix}\bar{w}_1\\ \bar{w}_2\end{bmatrix} + \begin{bmatrix}64/54\\ 0\end{bmatrix}\bar{u} \\
\bar{y} = [20/3\ ,\ 10]\begin{bmatrix}\bar{w}_1\\ \bar{w}_2\end{bmatrix}
\end{gathered} \tag{24}$$

SUMMARY, CONCLUSIONS, AND FUTURE RESEARCH

An nth order system can be transformed to a canonical representation that is related to the well-known Routh table in stability theory. This transformation causes the state vector to be partitioned into "slow" and "fast" variables. The Routh canonical representation is a function of n alpha coefficients (positive for a stable system) and n beta coefficients which are invariants of the system. The kth order Routh approximant is defined to be the kth order system whose alpha and beta coefficients are equal to the first k alpha and beta coefficients of the nth order system. Truncating the last (n-k) alpha and beta coefficients may be viewed as a singular perturbation method in which the magnitudes of $1/\hat{\alpha}_{k+1}, \ldots, 1/\hat{\alpha}_n$ are small and define the small parameter ϵ multiplying the derivative of the "fast" variables. Although the development in this paper is restricted to single input-single output systems that are asymptotically stable, the RAM extends to unstable and multiple input-multiple output systems. It also appears that the correlation between the RAM and singular perturbations extends to the more general case but the details remain to be worked out.

In defining the kth Routh approximant it was assumed that all the remaining (n-k) alpha coefficients are large relative to the first k alpha coefficients. It is possible for some of the remaining alpha coefficients to be small which could alter the partitioning of the state vector into "slow" and "fast" variables. This case does arise for a stable 4th order system with complex eigenvalues located roughly the same distance from the origin but with different real parts. This is a topic for further research but is a difficult problem to study analytically because the mapping between the alpha coefficients and the eigenvalues is nonlinear and complex.

The advantages of recognizing the correlation between the RAM and singular perturbations are twofold. First, since the RAM is a finite recursive algorithm readily programmed on a digital computer, it is a practical technique for applying singular perturbation theory to linear, time-invariant systems for which the small parameter(s) ϵ may not be explicitly known. Second, a valuable methodology for gaining insight and developing further properties of the RAM results from the application of singular perturbation theory. Of particular interest is the use of the RAM in the design of a control system for a large scale system.

REFERENCES

1. O'Malley, R. E., Jr., Introduction to Singular Perturbations, Academic Press, New York, 1974.
2. Kokotovic, P. V., "A Control Engineer's Introduction to Singular Perturbations," ASME publication Singular Perturbations: Order Reduction in Control System Design, presented at 1972 JACC, Stanford, California, August, 1972.
3. Kelley, H. J., "State Variable Selection and Singular Perturbations," ASME publication Singular Perturbations: Order Reduction in Control System Design, presented at 1972 JACC, Stanford, California, August, 1972.
4. Hutton, M. F., "Routh Approximation for High-Order Linear Systems," Proceedings Ninth Alleron Conference on Circuit and System Theory, Allerton House, Monticello, Illinois, October 6-8, 1971, pp. 160-169.
5. Hutton, M. F., "Routh Approximation Method for High-Order Linear Systems," Dissertation for Degree of Doctor of Philosophy (System Engineering) at Polytechnic Institute of New York, June 1974.
6. Hutton, M. F., and Friedland, B., "Routh Approximations for Reducing Order of Linear Systems," IEEE Transactions on Automatic Control, June 1975.
7. Hutton, M. F., and Rabins, M., "Simplification of High-Order Mechanical Systems Using the Routh Approximation," to appear in ASME Journal of Dynamic Systems, Measurement, and Control.

PROJECTIVE METHODS FOR REDUCING LINEAR DYNAMIC SYSTEMS AND APPLICATIONS TO ESTIMATION AND CONTROL

J. HICKIN and N.K. SINHA
Research Group in Simulation, Optimization and Control
Faculty of Engineering
McMaster University
Hamilton, Canada

ABSTRACT

It is shown that the method of linear system reduction by orthogonal projection may be adapted to fulfil the goal of braod applications under the unifying frame work of state aggregation. Applications include eigenvalue placement, near-optimal control, and reduced-order filtering.

INTRODUCTION

During the past decade several papers have been published on the subject of reduction of high-order linear dynamical systems [1-11]. The methods are roughly evenly divided between transfer function and state variable techniques, but little attention appears to have been given to applications. Indeed, the merit of most reduction methods is usually deduced from "how well" the output of the reduced system tracks that of the original high-order system. Unfortunately, good tracking of the output signal is no *a priori* guarantee that the reduced model is good for applications such as eigenvalue placement, near-optimal control or least-squares filtering.

Some qualitative results about these applications have been discussed by Mitra [4] in his work on reduction by orthogonal projection, and recently, Rao and Lamba [12] have shown that state feedback from the reduced-order model of Davison [1] can be used to place a subset of the eigenvalues of the large system arbitrarily. In this, they have been helped considerably by the concept of aggregation first introduced to control theory by Aoki [13].

An interesting insight is obtained by the observation that projective reduction is a special case of aggregation [14].

The object of this paper is to show how any projective method can be utilized for approximate placement of eigenvalues by output feedback based on the reduced model. Another application presented here is state estimation. This is not a reduced order Kalman filter to estimate the state of the large system, but rather a direct method for estimating the states of the reduced order system. The problem of near-optimal control by feeding back the state of the reduced-order model is also considered. Many other applications follow from the state estimation scheme presented.

Some Notes On Reduction By Orthogonal Projection [4]

We have a large scale system

$$\dot{x} = Ax + Bu \tag{1a}$$

$$y = Cx \tag{1b}$$

where $x \in R^n$, $u \in R^p$, and $y \in R^q$. For clarity, it is assumed that the eigenvalues of A are distinct and have negative real parts. Thus let

$$AV = V\Lambda \tag{2}$$

where

$$V \triangleq [v_1\ v_2\ \dots\ v_n] \tag{3a}$$

$$\Lambda \triangleq < \lambda_1, \lambda_2, \ldots, \lambda_n > \tag{3b}$$

Now assume that we wish to reduce the system order to $r < n$, retaining the first r eigenvalues. Partition V and Λ as follows

$$V = [V_{11} \; V_{12}] \tag{4a}$$

$$\Lambda = \begin{bmatrix} \Lambda_{11} & 0 \\ 0 & \Lambda_{22} \end{bmatrix} \tag{4b}$$

where

$$V_{11} = [v_1 \; v_2 \; \ldots \; v_r] \tag{5a}$$

$$V_{12} = [v_{r+1} \; v_{r+2} \; \ldots \; v_n] \tag{5b}$$

$$\Lambda_{11} = < \lambda_1, \lambda_2, \ldots, \lambda_r > \tag{5c}$$

$$\Lambda_{22} = < \lambda_{r+1}, \lambda_{r+2}, \ldots, \lambda_n > \tag{5d}$$

As shown by Mitra [4,6] we may reduce the system (1) as follows: Take

$$P = I_n - V_{12}(V_{12}^* W^{-1} V_{12})^{-1} V_{12}^* W^{-1} \tag{6a}$$

$$W = \int_0^\infty \exp(At) BDB^T \exp(A^T t) dt \tag{6b}$$

where D is any positive definite symmetric matrix. It is easily shown that (note from (6a) $PV_{12} = 0$, then use (2)-(5))

$$PA = PAP \tag{7}$$

Hence the "decontrolled" trajectory $\hat{x} = Px$ satisfies

$$\dot{\hat{x}} = PA\hat{x} + PBu \tag{8}$$

and

$$PA(PV_{11}) = (PV_{11}) \Lambda_{11} \tag{9}$$

Next, a matrix E is chosen whose r columns span the controllable subspace of system (8), so that

$$E = PV_{11}R \tag{10}$$

where R is any nonsingular $r \times r$ matrix. PA, PB, and E are partitioned as

$$PA = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix} \tag{11a}$$

$$PB = \begin{bmatrix} \hat{B}_{11} \\ \hat{B}_{21} \end{bmatrix} \tag{11b}$$

$$E = \begin{bmatrix} E_{11} \\ E_{21} \end{bmatrix} \tag{11c}$$

Mitra's formulae for the reduced system are then given by

$$F = E_{11}^{-1} \hat{A}_{11} E_{11} + E_{11}^{-1} \hat{A}_{12} E_{21} \tag{12a}$$

$$G = E_{11}^{-1} \hat{B}_{11} \tag{12b}$$

$$H = CE \tag{12c}$$

and

$$\dot{z} = Fz + Gu \tag{13a}$$

$$y_1 = Hz \tag{13b}$$

Finally in [14] it was shown that equations (12) may be rewritten as

$$F = KAE \tag{14a}$$

$$G = KB \tag{14b}$$

$$H = CE \tag{14c}$$

where the matrix K is given by

$$K = [E_{11}^{-1} \;\vdots\; 0]P \tag{15}$$

The Concept of Aggregation

Aoki [13] proposed the relationship

$$z = K_1 x \tag{16}$$

between the states of a system and any reduced order model. Such a matrix, if it exists, would satisfy

$$FK_1 = K_1 A \tag{17a}$$

$$G = K_1 B \tag{17b}$$

$$HK_1 = C \tag{17c}$$

and, in general, (17c) would actually be an approximation. It was shown in [15] that

$$FK = KA \tag{18}$$

for K given by (15). Thus, for projective reduction, K given by (15) is an aggregation matrix, and

$$z = Kx \tag{19}$$

This result is the keystone of all further developments.

Improvement of Output Tracking

It can be shown that

$$PK^n = E \tag{20}$$

$$F = KAE = KAPK^n = (KP)(AP)K^n = K(PA)K^n = KAK^n \tag{21}$$

where K^n is any right weak generalized inverse of K[16]. In particular, (21) holds for $K^n = K^+$, the pseudoinverse of K, which is unique [16]. Consider equation (14c), which may be viewed as a "solution" to

$$HK = C$$

In general, any "solution" for H will not be exact, but if we use the pseudoinverse K^+ and "solve" for H as (compare (14c))

$$H = CK^{+} \tag{22}$$

this will be a minimum (Euclidean) norm solution [16]. Then the output of the reduced system (13) for H given by (22) will be such as to minimize the least squares error between itself and the actual output of system (1).

Extension of Projective Reduction To Discrete Time Systems

In view of equations (18) and (21) it is not difficult to show that:

$$KA^{i} = F^{i}K \qquad , i = 0,\pm1,\pm2, \ldots \tag{23a}$$

$$F^{i} = KA^{i}K^{n} \qquad , i = 0,\pm1,\pm2, \ldots \tag{23b}$$

where K^{n} is any right weak generalized unverse of K. There immediately follows

$$K\exp(At) = \exp(Ft)K \tag{24a}$$

$$\exp(Ft) = K\exp(At)K^{n} \tag{24b}$$

Consider a discrete-time system obtained from system (1) where the sampling period is τ. Then

$$x_{k+1} = \phi_x x_k + \theta_x u_k \tag{25a}$$

$$y_k = C\, x_k \tag{25b}$$

where

$$\phi_x = \exp(A\tau) \tag{26a}$$

$$\theta_x = \int_0^{\tau} \exp(A\sigma)d\sigma B \tag{26b}$$

If the reduced system (13) is now discretized there results

$$z_{k+1} = \phi_z z_k + \theta_z u_k \tag{27a}$$

$$y_{1k} = Hz_k \tag{27b}$$

where by the use of equations (24), it is evident that

$$\phi_z = K\phi_x K^{n} \tag{28a}$$

$$\theta_z = K\theta_x \tag{28b}$$

thus K may also be used to reduce the discrete time counterpart of system (1). It is of interest to note that direct reduction of a discretized system is possible, but, in general, the projection cannot always be made orthogonal, for consider the analogue of (6a)

$$P_1 = I_n - V_{12}(V_{12}^{*} W_1^{-1} V_{12})^{-1} V_{12}^{*} W_1^{-1} \tag{29a}$$

The analogous formula for W_1 (viz 6b) would be

$$W_1 = \phi_x W_1 \phi_x^{T} + \theta_x D \theta_x^{T} \tag{29b}$$

Unfortunately, W_1 is not guaranteed to be positive definite, since the term, $\theta_x D\theta_x^T$ may be positive semi-definite [19]. This is an interesting parallel to a linear space incomplete in its norm [17]. In practice, however, good approximate results may be obtained by solving for W_1 as

$$W_1 = \phi_x W_1 \phi_x^T + \theta_x D\theta_x^T + \delta I_n \tag{30}$$

where δ is small in comparison with the eigenvalues of A. This forces W_1 to be positive definite, and hence W_1^{-1} will exist, and the use of (29a) is possible.

Applications

The reduced order model may be used for sub-optimal control of the large system, eigenvalue placement by state feedback, and approximate eigenvalue placement by output feedback. These ideas have been covered in [12-15,18], where Aoki [13] first proposed the use of aggregation for sub-optimal control, and Rao and Lamba [12] showed that state feedback from the reduced order model of Davison [1] could be used to place a subset of the eigenvalues of the large system arbitrarily. This idea was adapted into the more general setting of any projectively reduced model by Hickin and Sinha [15], and generalized to include approximate eigenvalue placement by output feedback, based on the reduced model. Another application which we present here, is state estimation. This is not a reduced order Kalman filter to estimate the state of the large system - this may be accomplished in a straightforward way by considering it as the dual to the sub-optimal regulator [13,18], rather we estimate directly the state of the reduced order system. It is felt that this is more in the spirit of applying model reduction.

The applications by no means end here, but to our knowledge, no further work has yet been published. Areas of further research include adaptive filtering to estimate the reduced model state, partial decoupling of the large scale system, zero assignment, etc. In fact, it should be possible to derive a "reduced analogue" for any procedure requiring the use of state variable feedback.

Sub-Optimal Control

This method transforms a quadratic performance index specified for the large system, into an "equivalent" one suitable to the reduced order model. The standard theory is then applied to the reduced order system with a great saving in computation. In [13] Aoki showed that if the quadratic performance index specified for the large system is

$$J = \int_o^T (x^T Qx + u^T Ru)dt \tag{31}$$

then its counterpart for the reduced model is given by

$$J_r = \int_o^T (z^T Q_r z + u^T Ru) \tag{32a}$$

where

$$Q_r = (K^+)^T QK^+ \tag{32b}$$

An heurisitic derivation of (32) follows from (19), viz, $x = K^+ z$, and substitution into (31). The resulting control law is sub-optimal, of course, but bounds on performance degradation may be calculated [13].

Eigenvalue Shifting By "Aggregated" State Feedback

The use of the reduced order model to derive state feedback laws to place the eigenvalues of the large system was first considered by Rao and Lamba [12] who used the reduced order model of Davison [1]. The subsequent generalization by Hickin and Sinha [15] is clearer, and more concise, and is given below. Assume that equations (2-5) hold.

Consider a feedback law

$$u = -\Gamma z = -\Gamma K x \tag{33}$$

Note that since $PV_{12} = 0$

$$(A-B\Gamma K)V_{12} = AV_{12} = V_{12}\Lambda_{22} \tag{34}$$

Let
$$(A-B\Gamma K)\bar{V}_{11} = \bar{V}_{11}\bar{\Lambda}_{11} \tag{35}$$

Then
$$K(A-B\Gamma K)\bar{V}_{11} = K\bar{V}_{11}\bar{\Lambda}_{11}$$

By (18), (14b) we get

$$(F-G\Gamma)K\bar{V}_{11} = K\bar{V}_{11}\bar{\Lambda}_{11} \tag{36}$$

Hence the eigenvalues of $F-G\Gamma$ are also eigenvalues of $A-B\Gamma K$. The remaining eigenvalues of $A-B\Gamma K$ remain unchanged, viz., (34). Thus, eigenvalues λ_i, $i = 1,2, \ldots, r$ may be arbitrarily placed by a feedback law of the form $u = -\Gamma Kx$ and these will also be eigenvalues of the reduced model with feedback law $u = -\Gamma z$.

Approximate Eigenvalue Shifting By Output Feedback [15]

The above idea is generalized to that of output feedback. Take $u_1 = -\Gamma Hz$, $u = -\Gamma Cx$. Now $u_1 \neq u$ as $HK = CK^+K \neq C$ in general. If, however,

$$||G\Gamma(I_n - K^+K)|| << \min_{1 \leq i \leq n} |\lambda_i| \tag{37}$$

the eigenvalues of $F-G\Gamma H$ will be approximate eigenvalues of $A-B\Gamma C$, and the remaining eigenvalues will correspond roughly to those of A.

Estimation of the Reduced Model State

The large and reduced models now become

$$\dot{x} = Ax + Bu \tag{38a}$$

$$y = Cx + v \tag{38b}$$

$$\dot{z} = Fz + Gu \tag{39a}$$

$$y_1 = Hz + v \tag{39b}$$

where

$$z = Kx \tag{40}$$

$$F = KAK^{+} \tag{41a}$$

$$G = KB \tag{41b}$$

$$H = CK^{+} \tag{41c}$$

and K^{+} is the pseudoinverse of K. Also the u and v are uncorrelated white noise sources,

$$E\{u(t)u^{T}(s)\} = Q\delta(t-s) \qquad E\{u(t)\} = 0 \tag{42a}$$

$$E\{v(t)v^{T}(s)\} = R\delta(t-s) \quad , \quad E\{u(t)\} = 0 \tag{42b}$$

A Kalman filter for the reduced order system would be given by (we assume that the reduced order model is observable)

$$\dot{\hat{z}} = F\hat{z} + \Gamma\eta_1 \quad , \quad \hat{z}(0) = 0 \tag{43a}$$

$$\eta_1 = y_1 - H\hat{z} \tag{43b}$$

where

$$\Gamma = PH^{T}R^{-1} \tag{44a}$$

$$P = E\{\tilde{z}(t)\tilde{z}^{T}(t)\} \quad ; \quad \tilde{z} \triangleq z-\hat{z} \tag{44b}$$

$$\dot{P} = FP + PF^{T} + GQG^{T} - \Gamma R\Gamma^{T} \tag{44c}$$

Such a filter cannot be implemented, however, as the observations process y_1 is fictitious. Note, however, that

$$y - y_1 = Cx - Hz = C(I_n-K^{+}K)x \triangleq C\phi x \tag{45}$$

Instead of y_1 we shall use y in the filter given by (43) and (44), and thus we expect a sub-optimal estimate, denoted by $\hat{z}_1$. Let

$$\hat{z}_1 = \hat{z} + \varepsilon \tag{46}$$

$$\dot{\hat{z}}_1 = F\hat{z}_1 + \Gamma\eta \quad , \quad \hat{z}_1(0) = 0 \tag{47a}$$

$$\eta = y - H\hat{z}_1 \tag{47b}$$

where Γ is given by equations (44). Use of (43), (45), and (46) gives

$$\dot{\varepsilon}(t) = (F-\Gamma H)\varepsilon(t) + \Gamma C\phi x \tag{48}$$

integrating and taking expectations gives

$$\begin{aligned} E\{\varepsilon(t)\} &= \phi(t,0)\varepsilon(0) + \int_{0}^{t} \phi(t,\tau)\Gamma(\tau)C\phi \; E\{x(\tau)\}d\tau \\ &= 0 + 0 = 0 \end{aligned} \tag{49}$$

where $\phi(t,\tau)$ is the state transition matrix of $F-\Gamma H$.

It is of interest to determine the difference between the estimate $K\hat{x}$ and $\hat{z}_1$. The full order Kalman filter is

$$\dot{\hat{x}} = A\hat{x} + \Gamma_x\eta_x \quad , \quad \hat{x}(0) = 0 \tag{50a}$$

$$\eta_x = y - C\hat{x} \tag{50b}$$

$$\Gamma_x = P_x C^T R^{-1} \tag{51a}$$

$$P_x = E\{\tilde{x}x^T\} \tag{51b}$$

$$\dot{P}_x = AP_x + P_x A^T - \Gamma_x R \Gamma_x^T + BQB^T \tag{51c}$$

It is easily seen that

$$(K\dot{P}_x K^T) = F(KP_X K^T) + (KP_X K^T)F^T - (K\Gamma_x)R(K\Gamma_x)^T + GQG^T \tag{52}$$

Comparison with equation (54c) indicates that $KP_x K^T$ corresponds to P and $K\Gamma_x$ to Γ. Note, however, that strict equality does not hold, as $HK = CK^+K \neq C$ in general. Hence some degradation in performance is unavoidable.

Conclusions

An applications oriented method of judging the merits of system reduction methods has been proposed and, at present, only projective methods fit this category. A number of applications of reduced models have been examined including state estimation, sub-optimal control, and eigenvalue placement. This by no means exhausts the possibilities, however, and further research should be profitable, for the state estimation scheme presented will allow the implementation of any algorithm requiring the state vector of the reduced order system. It is also evident, that the applicability of these techniques to discrete time systems is possible in a straightforward manner.

REFERENCES

[1] E.J. Davison, "A method for simplifying linear dynamic systems", IEEE Trans. AC-11, 1966, pp. 93-101.

[2] M.R. Chidambara and E.J. Davison, "On a method for simpligying linear dynamic systems", IEEE Trans. AC-12, 1967, pp. 119-120, pp. 213-214, 799-800.

[3] M.R. Chidambara, "Two simple techniques for the simplification of large dynamic systems", Proc. Joint Automatic Control Conf., 1969, pp. 699-674.

[4] D. Mitra, "On the reduction of complexity of linear, dynamical models", Technical report AEEW-R 520, United Kingdom Atomic Energy Authority, 1967.

[5] D. Mitra, "Analytical results on the use of reduced models in the control of linear dynamical systems", Proc. IEE-116, 1969, pp. 1439-1444.

[6] D. Mitra, "The reduction of complexity of linear, time invariant, dynamic systems", Proc. 4th IFAC, Technical Series 67, Warsaw, 1969, pp. 19-33.

[7] N.K. Sinha and W. Pille, "A new method for reduction of dynamic systems", Int. J. Control, vol. 14, 1971, pp. 111-118.

[8] N.K. Sinha and G.T. Bereznai, "Optimum approximation of high order systems by low order models", Int. J. Control, vol. 14, 1971, pp. 951-960.

[9] S.A. Marshall, "An approximate method for reducing the order of linear systems", Int. J. Control, vol. 10, 1966, pp. 642-643.

[10] J.H. Anderson, "Geometric approach to the reduction of dynamical systems", Proc. IEE-114, 1967, pp. 1014-1018.

[11] C.F. Chen and L.S. Shieh, "A novel approach to the problem of linear model simplification", Proc. Joint Automatic Control Conf., 1968, pp. 454-461.

[12] S. Vittal Rao and S.S. Lamba, "Eigenvalue assignment in linear-optimal control systems via reduced order models", Proc. IEE-122, 1975, pp. 197-201.

[13] M. Aoki, "Control of large scale dynamic system by aggregation", IEEE Trans. AC-13, 1968, pp. 246-253.

[14] J. Hickin and N.K. Sinha, "Aggregation matrices for a class of low-order models for large-scale systems", Electronics Letters, vol. 11, no. 9, May 1975, p. 186.

[15] J. Hickin and N.K. Sinha, "Some notes on eigenvalue assignment by reduced order models", Electronics Letters, vol. 11, July 1975, pp. 318-319.

[16] T.L. Boullion and P.L. Odell, "Generalized inverse matrices", Wiley-Interscience, New York, London, Sydney, and Toronto, 1971.

[17] A.W. Nalyor and G.W. Sell, "Linear operator theory in engineering and science", Holt, Rinehart, Winston, New York, 1971.

[18] S.S. Lamba and S. Vittal Rao, "Derivation of aggregation matrices for simplified models of linear dynamic systems and their application for optimal control", Proc. Joint Automatic Control Conf., 1972, pp. 498-503.

[19] K. Ogata, "State Space Analysis of Control Systems", Prentice-Hall, 1967.

TWO STAGE DESIGN OF SINGULARLY PERTURBED LINEAR REGULATORS

J. H. CHOW
University of Illinois, Urbana, Illinois

ABSTRACT

A decomposition method is proposed for the linear time-invariant singularly perturbed regulator problem. A fast subsystem and a slow subsystem are optimized with respect to their performance indices. A new form of near-optimal feedback is obtained and previous sufficiency conditions in singular perturbation theory are relaxed.

1. Introduction

In this paper we discuss the linear regulator design of a singularly perturbed system. The problem is the near-optimal control of the linear time-invariant singularly perturbed system

$$\dot{x}_1 = A_{11}x_1 + A_{12}x_2 + B_1 u, \ \dim x_1 = n_1, \ x_1(0) = x_{10} \tag{1a}$$

$$\mu\dot{x}_2 = A_{21}x_1 + A_{22}x_2 + B_2 u, \ \dim x_2 = n_2, \ x_2(0) = x_{20}, \ \mu \text{ scalar} \tag{1b}$$

$$y = C_1x_1 + C_2x_2 \tag{1c}$$

with respect to the performance index

$$J = 1/2 \int_0^\infty (y'y + u'Ru)dt \ , \ R > 0 \ . \tag{2}$$

This subject was first treated in [1,2]. In this paper the regulator design of the complete system is decomposed into two lower order problems. A fast subsystem and a slow subsystem are postulated and optimized with respect to their performance indices. A new form of near-optimal feedback is obtained and a set of sufficient conditions for its existence is relaxed.

2. Previous Results

In [2], a near optimal feedback control

$$u = F\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = -R^{-1}\begin{bmatrix} B_1 \\ B_2/\mu \end{bmatrix}'\begin{bmatrix} \hat{K}_1 & \mu K_2 \\ \mu K_2' & \mu K_3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3}$$

is designed in two stages. The full Riccati system for the regulator problem (1), (2) is formulated. At $\mu = 0$ the full Riccati system is decomposed into two smaller Riccati systems, one corresponding to x_2, and the other corresponding to x_1, and the x_2 Riccati system is independent of the x_1 Riccati system.

The designs of the two lower order Riccati gains are as follows. First, assuming that (A_{22}, B_2, C_2) is stabilizable-detectable (Condition a), the $n_2 \times n_2$ matrix K_3 is obtained from the x_2 Riccati system

$$0 = - K_3A_{22} - A'_{22}K_3 + K_3S_2K_3 - C'_2C_2 \quad , \; S_2 = B_2R^{-1}B'_2 \tag{4}$$

and the following matrices are evaluated

$$E_1 = (SK_3-A_{12})(A_{22}-S_2K_3)^{-1} \; , \; S = B_1R^{-1}B'_2 \tag{5a}$$

$$E_2 = (A'_{21}K_3+C'_1C_2)(A_{22}-S_2K_3)^{-1} \tag{5b}$$

$$\hat{A} = A_{11} + E_1A_{21} + SE'_2 + E_1S_2E'_2 \tag{6}$$

$$\hat{B} = B_1 + E_1B_2 \tag{7}$$

$$\hat{Q} = -E_2A_{21} - A'_{21}E'_2 - E_2S_2E'_2 + C'_1C_1 \equiv \hat{C}'\hat{C} \; . \tag{8}$$

Then the $n_1 \times n_1$ matrix $\hat{K}_1$ is obtained from the x_1 Riccati system

$$0 = -\hat{K}_1\hat{A} - \hat{A}'\hat{K}_1 + \hat{K}_1\hat{B}R^{-1}\hat{B}'\hat{K}_1 - \hat{C}'\hat{C} \tag{9}$$

and K_2 is calculated from

$$K_2 = \hat{K}_1E_1 - E_2 \quad . \tag{10}$$

For the solution of (9) it is assumed in [2] that $(\hat{A}, \hat{B}, \hat{C})$ is stabilizable-detectable (Condition b). Since E_1, E_2 in (5) and thus $\hat{A}$, $\hat{B}$ and $\hat{C}$ in (6)-(8) depend on K_3, it appears from [2] that the solution $\hat{K}_1$ of (9) depends on the solution K_3 of (4). Therefore following [2], it is uncertain whether for some K_3 Condition b can be violated and thus the first stage design may make the second stage impossible. However, this ambiguity has been eliminated in [3].

The new approach [3] is to decouple the regulator problem (1), (2) by separately optimizing a fast subsystem and a slow subsystem with respect to their performance indices. Since the designs for the two subsystems are completely independent, Condition b is relaxed.

3. Fast and Slow Subsystem Postulates

We begin by formulating the fast and slow subsystem regulator problems as in Problems 1 and 2. Then a set of sufficient conditions for the existence of solutions to these problems are stated in Lemmas 1 and 2.

To formulate the fast subsystem regulator problem, we first recognize that x_2 and u consist of slow parts, $\bar{x}_2$ and $\bar{u}$, and fast parts, $\tilde{x}_2$ and $\tilde{u}$, and

$$x_2 = \bar{x}_2 + \tilde{x}_2 \quad . \tag{11}$$

$$u = \bar{u} + \tilde{u} \; . \tag{12}$$

Since the fast parts decay rapidly, they can be expressed in a "stretched" time scale [4],

$$\tau = t/\mu . \tag{13}$$

With the above formulation, the fast subsystem and its performance index are postulated.

Problem 1

The regulator problem for the fast subsystem of (1), (2) is to optimize the system

$$\frac{d\tilde{x}_2(\tau)}{d\tau} = A_{22}\tilde{x}_2(\tau) + B_2\tilde{u}(\tau), \quad \tilde{x}_2(0) = x_{20} - \bar{x}_2(0) \tag{14}$$

with respect to the performance index

$$J_f = \mu/2 \int_0^\infty (\tilde{x}_2'(\tau)C_2'C_2\tilde{x}_2(\tau) + \tilde{u}'(\tau)R\tilde{u}(\tau))d\tau, \quad R > 0 . \tag{15}$$

To construct Problem 1 from the original problem, we obtain (14) from (1b) by considering only the fast parts of x_2 and u and observing that from (13)

$$\mu \frac{d}{dt} = \frac{d}{d\tau} . \tag{16}$$

J_f is obtained by considering only the penalty on the fast parts in (2)

$$J_f = 1/2 \int_0^\infty (\tilde{x}_2'(t)C_2'C_2\tilde{x}_2(t) + \tilde{u}'(t)R\tilde{u}(t))dt$$

$$= \mu/2 \int_0^\infty (\tilde{x}_2'(\tau)C_2'C_2\tilde{x}_2(\tau) + \tilde{u}'(\tau)R\tilde{u}(\tau))d\tau$$

by noting that from (13)

$$dt = \mu d\tau . \tag{17}$$

A set of sufficient conditions for the existence of the solution to the fast subsystem regulator problem and the optimal feedback are summarized in the following lemma.

Lemma 1

If (A_{22}, B_2, C_2) is stabilizable-detectable (Condition a), then the solution for the fast subsystem regulator problem (14), (15) exists and the optimal feedback is in the form

$$\tilde{u} = - R^{-1}B_2'K_3\tilde{x}_2 \tag{18}$$

where K_3 is the positive definite solution of (4).

Proof: If Condition a is satisfied, then the optimal feedback exists and is in the form

$$\tilde{u} = - \frac{1}{\mu} R^{-1} B_2'\bar{K}_3\tilde{x}_2$$

where $\bar{K}_3$ is the positive definite solution of

$$0 = - \bar{K}_3A_{22} - A_{22}'\bar{K}_3 + \frac{1}{\mu}\bar{K}_3S_2\bar{K}_3 - \mu C_2'C_2 .$$

Comparing to (4), $\overline{K}_3$ can be expressed as $\overline{K}_3 = \mu K_3$. Hence u can be written as (18) with K_3 the positive definite solution of (4).

To formulate the slow subsystem regulator problem, an approach similar to that in [3] is taken. The system (1) is partially closed by using the feedback

$$u = v + G_2(x_2 - w) \quad . \tag{19}$$

In (19), v is the new input and if A_{22}^{-1} exists, w is defined as

$$w = -A_{22}^{-1}(A_{21}x_1 + B_2 v) \tag{20}$$

such that letting $\mu \to 0$, $\overline{x}_2 = \overline{w}$. A bar above a state variable denotes its slow part. Furthermore, v and x_1, and hence w, can be considered as having slow parts only. G_2 is a design parameter and is permissible provided that $A_{22}+B_2G_2$ is nonsingular. Clearly, for optimal control of the fast subsystem, it is desirable that

$$G_2 = -R^{-1}B_2'K_3 \tag{21}$$

which can be obtained by considering the fast parts in (19). The slow part of u becomes

$$\overline{u} = \overline{v} \quad . \tag{22}$$

With the above development, the slow subsystem and its performance index are postulated.

Problem 2

The regulator problem for the slow subsystem of (1), (2) is to optimize the system

$$\dot{\overline{x}}_1 = A_o\overline{x}_1 + B_o\overline{v},\ A_o = A_{11} - A_{12}A_{22}^{-1}A_{21},\ B_o = B_1 - A_{12}A_{22}^{-1}B_2,\ \overline{x}_1(0) = \overline{x}_{10} = x_{10} \tag{23a}$$

$$\overline{y} = C_o\overline{x}_1 + D_o\overline{v},\ C_o = C_1 - C_2A_{22}^{-1}A_{21},\ D_o = -C_2A_{22}^{-1}B_2 \tag{23b}$$

where $x_1 \to \overline{x}_1$, $v \to \overline{v}$, $y \to \overline{y}$ at $\mu = 0$, with respect to the performance index

$$J_s = 1/2\int_0^\infty (\overline{y}'\overline{y} + \overline{v}'R\overline{v})dt,\ R > 0. \tag{24}$$

To construct Problem 2, we obtain (23) by feeding back (19) to (1) and letting $\mu = 0$. Then $x_1 \to \overline{x}_1$, $x_2 \to \overline{x}_2$, $y \to \overline{y}$, $v \to \overline{v}$ and $\overline{x}_2$ can be eliminated from the resulting equations. Also (24) is obtained from (2) by letting $y \to \overline{y}$, $u \to \overline{u}$ and $\overline{u} = \overline{v}$.

A set of sufficient conditions for the existence of the solution to the slow subsystem regulator problem and the optimal feedback are summarized in the following lemma.

Lemma 2

If (A_o, B_o, C_o) is stabilizable-detectable (Condition b_1), then the solution for the slow subsystem regulator problem (23), (24) exists and the optimal feedback is in the form of

$$\overline{v} = -(R+D_o'D_o)^{-1}(D_o'C_o+B_o'K_1)\overline{x}_1 \tag{25}$$

where K_1 is the positive definite solution of

$$\begin{aligned} 0 = &-K_1(A_o-B_o(R+D_o'D_o)^{-1}D_o'C_o) - (A_o-B_o(R+D_o'D_o)^{-1}D_o'C_o)'K_1 \\ &+ K_1B_o(R+D_o'D_o)^{-1}B_o'K_1 - (C_o'C_o - C_o'D_o(R+D_o'D_o)^{-1}D_o'C_o) \quad . \end{aligned} \tag{26}$$

Proof: Rewriting (24) in terms of $\overline{x}_1$ and $\overline{v}$, we obtain

$$J_s = 1/2\int_0^{\infty}(\overline{x}_1'C_o'C_o\overline{x}_1 + 2\overline{v}'D_o'C_o\overline{x}_1 + \overline{v}'(R+D_o'D_o)\overline{v})dt \quad . \tag{27}$$

From [5], the optimal feedback is in the form of (25) with K_1 satisfying (26) provided that $(A_o-B_o(R+D_o'D_o)^{-1}D_o'C_o,\ B_o,\ \hat{C}_o)$ is stabilizable-detectable, where $\hat{C}_o$ is defined as

$$\hat{C}_o'\hat{C}_o = C_o'C_o - C_o'D_o(R+D_o'D_o)^{-1}D_o'C_o \quad . \tag{28}$$

It is well known that $(A_o-B_o(R+D_o'D_o)^{-1}D_o'C_o,\ B_o)$ is stabilizable if and only if $(A_o,\ B_o)$ is stabilizable. Since $R > 0$, $I-D_o(R+D_o'D_o)^{-1}D_o' > 0$ and therefore is nonsingular. Define $\hat{C}_o'\hat{C}_o = C_o'M'MC_o$, where M is nonsingular and $M'M = I-D_o(R+D_o'D_o)^{-1}D_o'$. It follows immediately that $(A_o-B_o(R+D_o'D_o)^{-1}D_o'C_o, MC_o)$ is detectable if and only if $(A_o,\ C_o)$ is detectable. Thus the stabilizability-detectability of $(A_o,\ B_o,\ C_o)$ is sufficient to guarantee the existence of the stabilizing K_1 solution of (26).

Combining (18) and (25), we obtain the feedback control

$$\begin{aligned} u = &-[(I-R^{-1}B_2'K_3A_{22}^{-1}B_2)(R+D_o'D_o)^{-1}(D_o'C_o+B_o'K_1)+R^{-1}B_2'K_3A_{22}^{-1}A_{21}]x_1 \\ &- R^{-1}B_2'K_3x_2 \quad . \end{aligned} \tag{29}$$

In the next section, we shall show that (29) is the same as obtained in [2].

4. Separation of Designs

In this section, we first show that the K_1 obtained from (26) is the same as the $\hat{K}_1$ obtained from (9).

Theorem 1

The optimal Riccati gain, K_1, for the slow subsystem regulator problem is invariant for all designs of K_3. In particular, $K_1 = \hat{K}_1$.

Proof: The first part of the theorem can be proved easily by noting that K_3 does not appear in (26). The second part, which is shown in Appendix 1, can be proved by showing that

$$\hat{A} = A_o - B_o(R+D_o'D_o)^{-1}D_o'C_o \tag{30a}$$

$$\hat{B}R^{-1}\hat{B}' = B_o(R+D_o'D_o)^{-1}B_o' \tag{30b}$$

$$\hat{Q} = C_o'C_o-C_o'D_o(R+D_o'D_o)^{-1}D_o'C_o . \tag{30c}$$

In showing the equivalence in (30), the only assumptions are that

A_{22}^{-1} and $(A_{22}-B_2R^{-1}B_2'K_3)^{-1}$ exist and (4) holds. Since K_1 does not depend on K_3, we have separated the designs of K_1 and K_3. The following theorem utilizes the separated designs and shows that (29) is indeed a near-optimal feedback to the system (1), (2).

Theorem 2

If Conditions a and b_1 are satisfied, then the near-optimal solution to the regulator problem (1), (2) exists and the near-optimal feedback is in the form (29), where K_1 and K_3 are solutions to (26) and (4). Furthermore, there exists a positive μ^*, such that for $\mu \in (0,\mu^*]$ and $t \geq 0$, the near-optimal trajectories are

$$x_1(t) = \bar{x}_1(t) + O(\mu) \tag{31}$$

$$x_2(t) = \bar{x}_2(t) + \tilde{x}_2(\tau) + O(\mu) \tag{32}$$

$$\bar{x}_1(t) = e^{(A_o+B_oF_o)t}\bar{x}_1(0) \tag{33}$$

$$\bar{x}_2(t) = -A_{22}^{-1}(A_{21}+B_2F_o)\bar{x}_1(t) \tag{34}$$

$$F_o = -(R+D_o'D_o)^{-1}(D_o'C_o+B_o'K_1) \tag{35}$$

$$\tilde{x}_2(\tau) = e^{(A_{22}-S_2K_3)\tau}x_2(0), \quad \tau=t/\mu \tag{36}$$

$$\bar{x}_1(0) = x_1(0) = x_{10} \tag{37}$$

$$\tilde{x}_2(0) = x_2(0) - \bar{x}_2(0) = x_{20} - \bar{x}_2(0) \quad . \tag{38}$$

Proof: We shall show that the u in (29) is equivalent to the u in (3) evaluated at $\mu = 0$. Noting that

$$(I-R^{-1}B_2'K_3A_{22}^{-1}B_2)(R+D_o'D_o)^{-1} = R^{-1}(I+B_2'(A_{22}-S_2K_3)'^{-1}K_3B_2R^{-1})$$

(29) can be written as

$$\begin{aligned}
u = &- [R^{-1}(I+B_2'(A_{22}-S_2K_3)'^{-1}K_3B_2R^{-1})(-B_2'A_{22}'^{-1}C_2'(C_1-C_2A_{22}^{-1}A_{21}) \\
&+ (B_1'-B_2'A_{22}'^{-1}A_{12}')K_1)+R^{-1}B_2'K_3A_{22}^{-1}A_{21}]x_1- R^{-1}B_2'K_3\ x_2 \\
= &- R^{-1}B_1'K_1\ x_1- R^{-1}[B_2'(I+(A_{22}-S_2K_3)'^{-1}K_3S_2)(-A_{22}'^{-1}C_2'(C_1-C_2A_{22}^{-1}A_{21}) \\
&- A_{22}'^{-1}A_{12}'K_1)+B_2'(A_{22}-S_2K_3)'^{-1}K_3B_2R^{-1}B_1'K_1+B_2'K_3A_{22}^{-1}A_{21}]x_1- R^{-1}B_2'K_3\ x_2 \\
= &- R^{-1}B_1'K_1\ x_1- R^{-1}B_2'[(A_{22}-S_2K_3)'^{-1}(-(C_2'C_1+K_3A_{21}) + (K_3S-A_{12}')K_1)]x_1 \\
&- R^{-1}B_2'K_3\ x_2 \\
= &- R^{-1}B_1'K_1\ x_1- R^{-1}B_2'K_2'\ x_1 - R^{-1}B_2'K_3\ x_2 \\
= &- R^{-1}\begin{bmatrix} B_1 \\ B_2/\mu \end{bmatrix}'\begin{bmatrix} K_1 & \mu K_2 \\ \mu K_2' & \mu K_3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\Bigg|_{\mu=0}
\end{aligned} \tag{39}$$

Hence (29) is a near-optimal feedback as (3) is a near-optimal feedback to the regulator problem (1), (2).

The near-optimal trajectories (31) and (32) are the results of the approximation of the two stage design. (33) and (36) are the state responses to the subsystems (23) and (14) with the near-optimal feedback (29). (34) is obtained by substituting (25) into (20) and noting that $\overline{x}_2=\overline{w}$. (37) and (38) are the initial conditions for (23) and (14).

From (33), (34) and (36), we can conclude that the "inherent" responses of the slow and fast subsystems are invariant. Furthermore, K_1 and K_3 can be separately designed as in (26) and (4). Also, Condition b_1 is a major improvement over Condition b.

5. Example

Consider the magnetic tape control system

$$\begin{bmatrix} \dot{x}_1 \\ \mu\dot{x}_2 \end{bmatrix} = \left[\begin{array}{cc|cc} 0 & .4 & 0 & 0 \\ 0 & 0 & .345 & 0 \\ \hline 0 & -.524 & -.465 & .262 \\ 0 & 0 & 0 & -1.0 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1.0 \end{bmatrix} u$$

with the output

$$y = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

optimized with respect to

$$J = 1/2 \int_0^\infty (y'y + u'u)dt \quad .$$

Conditions a and b_1 are satisfied. From (4), K_3 is found to be

$$K_3 = \begin{bmatrix} 1.0403 & .18036 \\ .18036 & .046187 \end{bmatrix} .$$

From (26), K_1 is found to be

$$K_1 = \begin{bmatrix} 7.3840 & 5.9048 \\ 5.9048 & 7.1516 \end{bmatrix} .$$

Combining K_1 and K_3 to form K_2 (39) and putting $\mu = .1$,

$$K_{\text{near-optimal}} = \begin{bmatrix} 7.3840 & 5.9048 & .39931 & .10000 \\ 5.9048 & 7.1516 & .37997 & .086123 \\ .39931 & .37997 & .10403 & .018036 \\ .10000 & .086123 & .018036 & .0046187 \end{bmatrix} .$$

Comparing to the optimal K

$$K = \begin{bmatrix} 7.5400 & 6.1704 & .40534 & .10000 \\ 6.1704 & 7.4673 & .39510 & .089202 \\ .40534 & .39510 & .13044 & .024396 \\ .10000 & .089202 & .024396 & .0061996 \end{bmatrix}$$

the approximation is fairly good.

6. Conclusion

In this paper the near-optimal linear regular design of a singularly perturbed system is solved. The existence of the solution only requires the stabilizability and the detectability of the slow and the fast subsystems. A near-optimal control is formulated in terms of the mutually independent slow and fast Riccati gains. The main achievements of the paper are that not only are two lower order subsystems being solved, but we have separation of their designs.

7. Acknowledgments

The author wishes to express his gratitude to Prof. P. V. Kokotovic of the University of Illinois, Urbana, for his suggestion and guidance in the course of this research. This work was supported in part by the National Science Foundation under Grant ENG 74-20091, in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-72-C-0259, and in part by the U.S. Air Force under Grant AFOSR 73-2570.

APPENDIX 1: PROOF OF SECOND PART OF THEOREM 1

It is desired to show that the equations in (30) are true. Define

$$H = I+R^{-1}B_2'K_3(A_{22}-B_2R^{-1}B_2'K_3)^{-1}B_2 \tag{A1}$$

$$H^{-1} = I-R^{-1}B_2'K_3A_{22}^{-1}B_2 \ .$$

H and H^{-1} are well-defined provided that A_{22} and $A_{22}-B_2R^{-1}B_2'K_3$ are nonsingular. Using (4)

$$\begin{aligned} H'^{-1}RH^{-1} &= R+B_2'A_{22}'^{-1}(-K_3A_{22}-A_{22}'K_3-K_3S_2K_3)A_{22}^{-1}B_2 \\ &= R+D_o'D_o \ . \end{aligned}$$

From (7) and (A1)

$$\begin{aligned} \hat{B}H^{-1} &= B_1-SK_3A_{22}^{-1}B_1+E_1(I-S_2K_3A_{22}^{-1})B_2 \\ &= B_o \ . \end{aligned}$$

Then

$$\begin{aligned} \hat{B}R^{-1}\hat{B}' &= (\hat{B}H^{-1})HR^{-1}H'(\hat{B}H^{-1})' \\ &= B_o(R+D_o'D_o)^{-1}B_o' \ . \end{aligned}$$

Similarly

$$\begin{aligned} \hat{A} &= A_{11}+E_1A_{21}+(B_1+E_1B_2)R^{-1}B_2'E_2' \\ &= A_{11}-(A_{12}-SK_3-E_1S_2K_3)A_{22}^{-1}A_{21}+(B_1+E_1B_2)R^{-1}B_2'E_2' \\ &= A_{11}-A_{12}A_{22}^{-1}A_{21}+(B_1+E_1B_2)R^{-1}B_2'(A_{22}-S_2K_3)'^{-1}(A_{22}'-K_3S_2)K_3A_{22}^{-1}A_{21} \\ &\qquad + (B_1+E_1B_2)R^{-1}B_2'(A_{22}-S_2K_3)'^{-1}(C_2'C_1+K_3A_{21}) \\ &= A_o+\hat{B}R^{-1}B_2'(A_{22}-S_2K_3)'^{-1}(C_2'C_1+(K_3A_{22}+A_{22}'K_3-K_3S_2K_3)A_{22}^{-1}A_{21}) \\ &= A_o+\hat{B}R^{-1}B_2'(A_{22}-B_2R^{-1}B_2'K_3)'^{-1}A_{22}'A_{22}'^{-1}C_2'(C_1-C_2A_{22}^{-1}A_{21}) \\ &= A_o+\hat{B}R^{-1}H'B_2'A_{22}'^{-1}C_2'C_o \\ &= A_o-B_o(R+D_o'D_o)^{-1}D_o'C_o \end{aligned}$$

$$\begin{aligned} \hat{Q} = C_1'C_1&-(C_1'C_2+A_{21}'K_3)(I+(A_{22}-S_2K_3)^{-1}S_2K_3)A_{22}^{-1}A_{21} \\ &-A_{21}'A_{22}'^{-1}(I+K_3S_2(A_{22}-S_2K_3)'^{-1})(C_2'C_1+K_3A_{21}) \\ &-(C_1'C_2+A_{21}'K_3)(A_{22}-S_2K_3)^{-1}B_2R^{-1}B_2'(A_{22}-S_2K_3)'^{-1}(C_2'C_1+K_3A_{21}) \end{aligned}$$

$$+A'_{21}{A'_{22}}^{-1}K_3S_2K_3A_{22}^{-1}A_{21}-A'_{21}{A'_{22}}^{-1}K_3S_2K_3A_{22}^{-1}A_{21}$$

$$= C'_1C_1-A'_{21}{A'_{22}}^{-1}(C'_2C_1+K_3A_{21})-(C'_1C_2+A'_{21}K_3)A_{22}^{-1}A_{21}+A'_{21}{A'_{22}}^{-1}K_3S_2K_3A_{22}^{-1}A_{21}$$

$$-[C'_1C_2+A'_{21}K_3+A'_{21}{A'_{22}}^{-1}K_3(A_{22}-S_2K_3)](A_{22}-S_2K_3)^{-1}B_2R^{-1}B'_2(A_{22}-S_2K_3)'^{-1}$$

$$\cdot[C'_2C_1+K_3A_{21}+(A'_{22}-K_3S_2)K_3A_{22}^{-1}A_{21}]$$

$$= C'_1C_1-A'_{21}{A'_{22}}^{-1}C'_2C_1-C'_1C_2A_{22}^{-1}A_{21}+A'_{21}{A'_{22}}^{-1}(-K_3A_{22}-A'_{22}K_3+K_3S_2K_3)A_{22}^{-1}A_{21}$$

$$-[C'_1C_2-A'_{21}{A'_{22}}^{-1}(-K_3A_{22}-A'_{22}K_3+K_3S_2K_3)]A_{22}^{-1}B_2(I+R^{-1}B'_2K_3(A_{22}-S_2K_3)^{-1}B_2)$$

$$\cdot R^{-1}(I+B'_2(A_{22}-S_2K_3)'^{-1}K_3B_2R^{-1})B'_2{A'_{22}}^{-1}[C'_2C_1-(-K_3A_{22}-A'_{22}K_3+K_3S_2K_3)A_{22}^{-1}A_{21}]$$

$$= C'_oC_o-C'_oD_o(R+D'_oD_o)^{-1}D'_oC_o \quad .$$

Hence $\hat{K}_1 = K_1$.

LIST OF REFERENCES

[1] P. Sannuti and P. V. Kokotovic, "Near-Optimal Design of Linear System by a Singular Perturbation Method," IEEE Trans. on Automatic Control, Vol. AC-14, No. 1, Feb. 1969, pp. 15-21.

[2] P. V. Kokotovic and R. A. Yackel, "Singular Perturbation of Linear Regulators: Basic Theorems," IEEE Trans. on Automatic Control, Vol. AC-17, Feb. 1972, pp. 29-37.

[3] J. H. Chow, "Separation of Time Scales in Linear Time-Invariant Systems," M. S. Thesis, Department of Electrical Engineering, University of Illinois, Urbana, Sept. 1975.

[4] P. V. Kokotovic, R. E. O'Malley, Jr., and P. Sannuti, "Singular Perturbations and Order Reduction in Control Theory--An Overview," 6th IFAC Congress, Preprint Part 1C, Aug. 1975, Boston.

[5] B. D. O. Anderson and J. B. Moore, Linear Optimal Control, Prentice-Hall, 1971.

AN ITERATIVE ALGORITHM FOR DETERMINING A TRANSFER FUNCTION FROM AMPLITUDE FREQUENCY RESPONSE DATA*

M. T. JONG AND K. S. SHANMUGAM
Department of Electrical Engineering
Wichita State University
Wichita, Kansas 67208

SUMMARY

The various methods existing in the literature for determining a transfer function from frequency response data require either the real and imaginary parts or the amplitude and phase characteristics of the frequency response. In this paper an iterative algorithm is presented for finding a transfer function from the amplitude response data alone. We are given the amplitude data D_k at the frequency $\omega=\omega_k$, $k=1,2,\ldots,L$. From this data set we wish to determine the transfer function as the ratio of two polynomials in the complex Laplace variable s. We start from the magnitude-squared function $G(\omega^2)=|F(j\omega)|^2=P(\omega^2)/Q(\omega^2)$, where $P(\omega^2) = \sum_{j=0}^{M} a_j\omega^{2j}$ and $Q(\omega^2) = 1 + \sum_{j=1}^{N} b_j\omega^{2j}$. Let P_{k_t}, Q_{k_t}, G_{k_t}, a_{j_t}, and b_{j_t} refer to the values of $P(\omega_k^2)$, $Q(\omega_k^2)$, $G(\omega_k^2)$, a_j, and b_j, respectively, at the end of the t^{th} iteration. With the weight during the t^{th} iteration set equal to the ratio $Q_{k_t}/Q_{k_{t-1}}$, the sum of the weighted squared errors is

$$E_t^{M,N} = \sum_{k=1}^{L} [Q_{k_t}(G_{k_t} - D_k^2)/Q_{k_{t-1}}]^2 = \sum_{k=1}^{L} (P_{k_t} - D_k^2 Q_{k_t})^2/Q_{k_{t-1}}^2$$

Setting the partial derivatives of $E_t^{M,N}$ to zero with respect to the a_{j_t}'s and b_{j_t}'s yields,

$$\sum_{j=0}^{M} [\sum_{k=1}^{L} \omega_k^{2(i+j)}/Q_{k_{t-1}}^2] a_{j_t} - \sum_{j=1}^{N} [\sum_{k=1}^{L} D_k^2\omega_k^{2(i+j)}/Q_{k_{t-1}}^2] b_{j_t} = \sum_{k=1}^{L} D_k^2\omega_k^{2i}/Q_{k_{t-1}}^2 \qquad i=0,1,\ldots,M,$$

$$\sum_{j=0}^{M} [\sum_{k=1}^{L} D_k^2\omega_k^{2(i+j)}/Q_{k_{t-1}}^2] a_{j_t} - \sum_{j=1}^{N} [\sum_{k=1}^{L} D_k^4\omega_k^{2(i+j)}/Q_{k_{t-1}}^2] b_{j_t} = \sum_{k=1}^{L} D_k^4\omega_k^{2i}/Q_{k_{t-1}}^2 \qquad i=1,2,\ldots,N.$$

This set of linear equations can be solved iteratively for the a_j's and b_j's until the solution converges according to some criterion. Then from the magnitude-squared function $G(\omega^2)$ a minimum-phase transfer function can be found. This technique has been successfully applied to system reduction and system identification. In system reduction, a higher-order transfer function is to be approximated by a lower-order one in the frequency range of interest. In this case, the degrees M and N are fixed. In system identification, a transfer function is to be found which best fits the amplitude data in the least-squares sense.

---

*This work was supported by RADC Contract No. AF30602-75-C-0121/SU-1229-WSU, and by NSF Grant No. Eng-74-18981.

TIME DOMAIN APPROXIMATION USING A DAMPED LEAST-SQUARES ALGORITHM

W.S. ISHAK and N.K. SINHA
Research Group in Simulation, Optimization
and Control, Department of Electrical Engineering,
McMaster University, Hamilton, Canada

ABSTRACT

The method of damped least squares is applied to the solution of problems of time-domain approximation, system-modeling and data fitting. The algorithm is based on the recent work of Massara and Fidler and the method of Levenberg and Marquardt. A number of examples illustrate the method.

INTRODUCTION

The problem of time-domain approximation is well-known in Circuit and System Theory, and may be stated simply as the determination of a rational transfer function of a specified order, the inverse Laplace transform of which closely approximates a given function of time. Different techniques have been applied to solve the problem stated above Chatterjee and Fahmy [1] and Joseph and Huang [2], [3] have recently applied elegant techniques, basbased on the orthogonal projection theorem in functional analysis, to determine a finite sum of real exponentials to approximate the given function such that the integral square error is minimized. The methods proposed by Chatterjee and Fahmy, as well as those proposed by Joseph and Huang are iterative techniques which start from a one-pole approximation and successively optimize residues and poles.

Sinha, Sen and To [4] have applied two digital methods for time-domain approximation. The first is a direct approach, based on the recursive algorithm for calculating the pseudo-inverse of a matrix [5]. The second method is based on the application of gradient search techniques to determine the parameters of the continuous-time system transfer function. The two methods have been applied to a number of examples and considerable improvements over Joseph and Huang technique have been observed.

It is the purpose of this paper to describe the application of a recent damped least-squares algorithm to solve time-domain approximation problems. The new algorithm was suggested by Massara and Fidler [6] and implemented and tested by Ishak [7].

STATEMENT OF THE PROBLEM

It is desired to find a function g(t), the Laplace transform, G(s), of which is a rational function of a specified order so that the following cost function is minimized:

$$J_p = \sum_{k=0}^{N} \{f(kT) - g(kT)\}^p \qquad (1)$$

where f(t) is the given function and p is a specified even integer which will be taken as 2.

THE DAMPED LEAST-SQUARES

In its simplest form, the least-squares minimization problem can be stated as follows:

$$\text{minimize} \qquad U = \sum_{i=1}^{N} \{ e_i (\underset{\sim}{\phi}) \}^2 \qquad (2)$$

where $$\underset{\sim}{e}(\underset{\sim}{\phi}) \equiv [e_1\ e_2\ \dots\ e_N]^T \tag{3}$$

is the vector of residuals,

and $$\underset{\sim}{\phi} = [\phi_1\ \phi_2\ \dots\ \phi_k]^T \tag{4}$$

is the independent variable vector.

The Gauss' least-squares algorithms proceed by obtaining the correction vector $\underset{\sim\sim}{\Delta\phi}$ in the form:

$$\underset{\sim\sim}{\Delta\phi} = -(\underset{\sim}{J}^T\ \underset{\sim}{J})^{-1}\ \underset{\sim}{J}^T\ \underset{\sim}{e} \tag{5}$$

where $\underset{\sim}{J}$ = Jacobian matrix defined as:

$$\underset{\sim}{J} = \begin{bmatrix} \frac{\partial e_1}{\partial \phi} & \frac{\partial e_1}{\partial \phi_2} & \cdot\ \cdot\ \cdot & \frac{\partial e_1}{\partial \phi_k} \\ \frac{\partial e_2}{\partial \phi_1} & \frac{\partial e_2}{\partial \phi_2} & \cdot\ \cdot\ \cdot & \frac{\partial e_2}{\partial \phi_k} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ \frac{\partial e_N}{\partial \phi_1} & \frac{\partial e_N}{\partial \phi_2} & \cdot\ \cdot\ \cdot & \frac{\partial e_N}{\partial \phi_k} \end{bmatrix} \tag{6}$$

In order to overcome the difficulties arising from large predicted $\underset{\sim\sim}{\Delta\phi}$ which causes erratic moves as the optimization proceeds, Levenberg [8] and Marquardt [9] suggested using damping methods which restrict the size of the step $\underset{\sim\sim}{\Delta\phi}$. Marquardt-Levenberg algorithms lead to a predicted correction of:

$$\underset{\sim\sim}{\Delta\phi} = -\ (\underset{\sim}{J}^T\underset{\sim}{J} + \lambda\underset{\sim}{I})^{-1}\ \underset{\sim}{J}^T\ \underset{\sim}{e} \tag{7}$$

with the positive weighting factor λ, termed the Levenberg parameter, and $\underset{\sim}{I}$ the unit matrix.

In the recent algorithm of damped least-squares the parameter λ is used efficiently to control the minimization process. λ is used to restrict the matrix of coefficients $(\underset{\sim}{J}^T\underset{\sim}{J} + \lambda\underset{\sim}{I})$ to be positive-definite and non-singular. The first property assures that $\underset{\sim\sim}{\Delta\phi}$ will be downhill and the second property avoids divergent iterations.

The algorithm has been tested for a variety of problems and very good results were obtained.

NUMERICAL EXAMPLES

Since the search directions in the damped least-squares require the evaluations of the partial derivatives of the cost function with respect to the variable parameters, expressions for these derivatives will now be derived.

Let

$$g(t) = \sum_{i=1}^{n} A_i \exp(-\alpha_i t) \tag{8}$$

represent the time-domain approximation to the given function, f(t). The cost function, defined in eqn. (1) may now be written, with p = 2, as:

$$J = \sum_{k=0}^{N} \left[f(kT) - \sum_{i=1}^{n} A_i \exp(-\alpha_i kT) \right]^2 \qquad (9)$$

we have:

$$\frac{\partial J}{\partial A_i} = \sum_{k=0}^{N} -2[f(kT) - g(kT)] \frac{\partial g(kT)}{\partial A_i} \qquad (10)$$

and

$$\frac{\partial J}{\partial \alpha_i} = \sum_{k=0}^{N} -2[f(kT) - g(kT)] \frac{\partial g(kT)}{\partial \alpha_i} \qquad (11)$$

where:

$$\frac{\partial g(kT)}{\partial A_i} = \exp(-\alpha_i kT) \qquad (12)$$

and

$$\frac{\partial g(kT)}{\partial \alpha_i} = -kTA_i \exp(-\alpha_i kT) \qquad (13)$$

The first example considered is one of the two discussed by Joseph and Huang [2], [3]. The function to be approximated is

$$f(t) = \exp(-0.5t) \cos t \qquad (14)$$

with a rational function G(s) having poles only on the negative real axis. The approximating function with four real poles, for damped least-squares approximation was determined, using the new Massara-Fidler algorithm, over the time interval 0 to 10 seconds, with samples at the interval of 0.05 seconds. It is given by:

$$g(t) = 523.9812 \exp(-0.776078\ t) - 203.9093 \exp(-1.061022\ t) + 868.1805 \exp(-0.9149355\ t) - 1187.324 \exp(-0.8294795\ t) \qquad (15)$$

This result may be compared with the result of Joseph and Huang, given as

$$g_1(t) = 30.94897 \exp(-3.91206\ t) - 92.5749 \exp(-5.33332\ t) + 82.85118 \exp(-6.63192\ t) - 20.30795 \exp(-8.90253\ t) \qquad (16)$$

In Fig. 1 is shown a plot of f(t), g(t), $g_1(t)$ from which it is easy to conclude that g(t) is a much better approximation to f(t).

A quantitative comparison can be obtained by comparing the minimum sum of squares over the interval 0 to 10 seconds, with a sampling interval of 0.05 seconds. These values are given as:

$$\sum_{k=0}^{200} \{f(kT) - g(kT)\}^2 = 0.1343226 \qquad (17)$$

$$\sum_{k=0}^{200} \{f(kT) - g_1(kT)\}^2 = 1.752 \qquad (18)$$

It should be mentioned here that this example has been tried before by Sinha, et al [4] and the approximating function, using the Fletcher method, was:

$$g_2(t) = 524 \exp(-0.761504\ t) - 203.89 \exp(-1.035\ t) + 868.199 \exp(-0.89579\ t) - 1187.309 \exp(-0.813324\ t) \qquad (19)$$

with

$$\sum_{k=0}^{200} [f(kT) - g_2(kT)]^2 = 0.149185 \qquad (20)$$

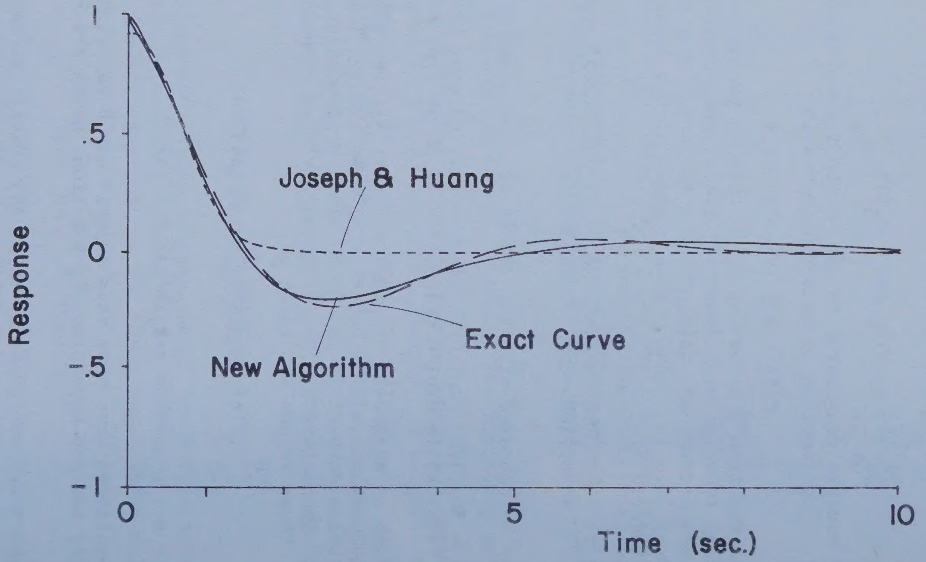


Fig. 1 Approximation to exp(-.5t) cost

It is clear that the values obtained for the independent parameters using the damped least-squares method (eqn. (15) and those obtained using Fletcher method (eqn. (19)) are very close, but the minimum sum of squares obtained using the first method is less than that obtained using the second method.

The second example is to find a second order-model the impulse response of which approximates that of a given fourth-order system (Bandler, et. al., [10]), in the least-squares sense.

The transfer function of the fourth-order system is:

$$G(s) = \frac{(s+4)}{(s+1)(s^2+4s+8)(s+5)} \tag{21}$$

and the transfer function of the second-order model is

$$H(s) = \frac{x_3}{(s+x_1)^2+x_2^2} \quad . \tag{22}$$

The problem is therefore equivalent to finding the optimum point $\underset{\sim}{x}$ such that the function:

$$F(\underset{\sim}{x},t) = \frac{x_3}{x_2} \exp(-x_1 t) \sin x_2 t \tag{23}$$

is the best approximation to the function

$$g(t) = \frac{3}{20} \exp(-t) + \frac{1}{52} \exp(-5t) - \frac{\exp(-2t)}{65} (3 \sin 2t + 11 \cos 2t) \tag{24}$$

in the least-squares sense. The problem was discretized into 51 uniformly spaced points in the interval 0 to 10 seconds and the function to be minimized is given by:

$$U = \sum_{i=1}^{51} e_i^2(\underset{\sim}{x}) \tag{25}$$

where:

$$e_i(\underset{\sim}{x}) = F(\underset{\sim}{x}, t_i) - s(t_i). \tag{26}$$

The results obtained are:

$$U = 4.3682967 \times 10^{-4} \tag{27}$$

and

$$x = \begin{bmatrix} 1.0164706 \\ 0.7892702 \\ 0.1614000 \end{bmatrix} \tag{28}$$

CONCLUSION

The recent algorithm for damped least-squares problems has been used efficiently for solving the time-domain approximation problem. Several examples have been used, indicating improvements over previously tried methods.

ACKNOWLEDGEMENT

The authors would like to acknowledge the very helpful discussions with Dr. J.W. Bandler.

REFERENCES

[1] R.N. Chatterjee and M.M. Fahmy, "On Time-Domain Approximation", Proc. of the 11th Midwest Symposium on Circuit Theory, May 1968, pp. 288-298.

[2] R.D. Joseph and S.L. Huang, "Iterative Method of Real Exponential Approximation", Proc., of the 13th Midwest Symposium on Circuit Theory, May 1970, pp. IV 2.1-2.6.

[3] R.D. Joseph and S.L. Huang, "Approximation in the Time Domain", Proc. of the 14th Midwest Symposium on Circuit Theory, May 1971, pp. 15.2.1-15.2.10

[4] N.K. Sinha, A. Sen and R.H.Y. To, "Time Domain Approximation Using Digital Methods, Int. J. Syst. Sci., Vol.5, No.4, 1974, pp. 373-382.

[5] N.K. Sinha and W. Pille, "Online Parameter Estimation Using Matrix Pseudoinverse", Proc. IEE, Vol. 118, No. 8, August 1971, pp. 1041-1046.

[6] R.E. Massara and J.K. Fidler, "Efficient Damping Method for Least-Squares Algorithms", Elect. Lett., Vol. 11, No. 2, Jan. 75, pp. 33-34.

[7] W. Ishak, "SQUARE - An Implementation of the Massara - Fidler Algorithm for Least-Squares Problems", Internal Report, No. SOC-88, Dept. of Elect. Engrg., McMaster University, Hamilton, Ontario, Canada, 1975.

[8] K. Levenberg, "A Method for the Solution of Certain Non-Linear Problems in Least Squares", Quart. J. Appl. Math., Vol. 2, 1944, pp. 164-168.

[9] D.W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters", SIAM J., Vol. 11, No. 2, June 1963, pp. 431-441.

[10] J.W. Bandler, C. Charalambous and J.H.K. Chen, "MINOPT - An Optimization Program Based On Recent Minimax Results", Internal Report, No. SOC-70, Dept. Elect. Engrg., McMaster University, Hamilton, Ontario, Canada, 1974.

BOUNDS AND BOUNDING FUNCTIONS FOR A STATIONARY GI/G/1 QUEUE

S. CALO
RCA David Sarnoff Res. Ctr.
Princeton, NJ 08544

S. Schwartz
Dept. of Electrical Eng.
Princeton University
Princeton, NJ 08540

ABSTRACT

New bounds for the equilibrium mean wait in a standard (GI/G/1) queue are obtained. An upper bound is shown to be a more accurate approximation at low-to-moderate traffic intensities than the Kingman upper bound and a new lower bound is shown to consistently track the true mean better than the Marshall lower bound.

INTRODUCTION

Two new bounds are exhibited for the equilibrium mean wait in a stationary GI/G/1 queue and an alternate proof of Marshall's lower bound is given. The new lower bound is shown to dominate the Kingman lower bound, and, in the examples investigated, consistently tracks the true mean better than the Marshall bound. Examples of the new upper bound show it to be a more accurate approximation at low-to-moderate traffic intensities than the Kingman upper bound, while not differing from it significantly at high levels of traffic. The computation of these bounds involves the marginal overload distribution.

In the next section we establish notation and derive a lower bound, called the expectation lower bound (Eq. (2.13)). In section 3 we obtain companion "intersection lower and upper bounds." The lower bound was first derived by Marshall and the upper bound appears to be new. These bounds are evaluated and compared in Section 4.

PRELIMINARIES

A single server queueing process is specified by two random sequences: $\{\tau_n\}_{n=1}^{\infty}$, the time intervals between arrivals of successive customers; and $\{s_n\}_{n=1}^{\infty}$, their required service times. In a standard queue, it is assumed that these sequences consist of i.i.d. random variables and that they are independent of one another. Designating the marginal overloads $\{X_n\}_{n=1}^{\infty}$ by $X_n = S_n - \tau_n$ and considering a first-come-first-served (FCFS) priority discipline, the waiting time sequence $\{w_n\}_{n=1}^{\infty}$ is given by the recursion:

$$\begin{aligned} w_1 &= 0 \\ w_{n+1} &= f(w_n + X_n), \quad n \geq 1 \end{aligned} \tag{2.1}$$

where $f(y) = \max(0,y) = (y)^+ = \frac{y+|y|}{2}$ is a non-negative continuous and monotone non-decreasing function.

Throughout, we shall make the following assumptions:

a) All random variables are second order: $\mathrm{Var}(\tau) < \infty, \mathrm{Var}(S) < \infty$, which implies that $E(\tau) = 1/\lambda < \infty$, and $E(S) = 1/\mu < \infty$. b) The traffic intensity is less than one: $\rho = E(S)/E(\tau) = \lambda/\mu < 1$. Under this stability assumption ($\rho < 1$), it is known that w_n converges in distribution to an a.s. finite r.v. w' and the steady-state behavior of the queue can be expressed as (see[1] for the details)

$$w' = f(w + X) \tag{2.2}$$

where the r.v. w' has the same distribution as w.

The idle time is defined by $d_n = f[-(w_n + X_n)]$ which converges in distribution to :

$$d_\infty = f[-(w + X)] \tag{2.3}$$

We make use of the stationarity of the sequence and the identities

$$\begin{aligned} Y &= f(Y) - f(-Y) \\ \mathrm{Var}\ (Y) &= \mathrm{Var}(f(Y) - f(-Y)) \qquad (2.4) \\ &= \mathrm{Var}\ f(Y) + \mathrm{Var} f(-Y) - 2\mathrm{Cov}[f(Y), f(-Y)] \end{aligned}$$

to obtain the mean of the equilibrium idle time

$$\bar{d}_\infty = E(-X)$$

and the variance of the equilibrium wait

$$\mathrm{Var}(w+X) = \mathrm{Var}(f(w+X)) + \mathrm{Var}(d_\infty) + 2\ E(w')E(-X)$$

Let $W_D = 1/(2\mu^2 E(-X)) = \rho/(2\mu(1-\rho))$ denote the expected wait for an M/D/1 queue. The above two moment expressions are combined to obtain a simple formula for the expected equilibrium wait in terms of the idle time:

$$\begin{aligned} Ew' = Ew &= \mu^2 W_D[\mathrm{Var}(X) - \mathrm{Var}(d_\infty)] \\ &= \mu^2 W_D[E(X^2) - E(d_\infty^2)] \end{aligned} \tag{2.5}$$

The second expression follows from the stability assumption, i.e., with $\rho < 1$, $E^2 X = E^2 d_\infty$.

Two bounds are easily obtained by approximating the idle time variance. Since $\mathrm{Var}(d_\infty) \geq 0$, we have

$$Ew' = Ew \leq \mu^2 W_D\ \mathrm{Var}(X), \tag{2.6}$$

which is Kingman's well-known upper bound [2]. From the inequality $f[-(w+X)] \leq f(-X)$, we have $Ed_\infty^2 = Ef^2[-(w+X)] \leq Ef^2(-X) = EX^2 - Ef^2(X)$, and Kingman's lower bound follows

$$Ew \geq \mu^2 W_D E f^2(X) \tag{2.7}$$

Under conditions of high traffic intensity, the upper bound is a good approximation to the mean wait. It has the added advantage of depending only on first and second moments. The development of bounds for low-to-moderate traffic intensities leads to somewhat more involved expressions. The bounds to be presented below will involve straightforward operations on the overload distribution function, $\Pr\{X \leq x\} = G(x)$. In order to derive these bounds, two alternate representations for the equilibrium mean wait must first be obtained.

We will use conditional expectations with the notation E_Y^Z

to designate the expectation of the random variable y conditioned on the variable z. When it is clear from the context, E alone will denote expectation (unconditional) of the indicated random variables.

From Eq. (2.2), the first representation is:

$$\begin{aligned} Ew' &= Ew = Ef(w + X) \\ &= E_w\{E_x^w f(w + X)\} = E_w g(w) \end{aligned} \tag{2.8}$$

where we have defined the lower bounding function

$$g(y) = E_x^y\, f(y + X) = \int_x f(y + x)\, dG(x) \tag{2.9}$$

The second line follows from the independence of w and X, i.e., from the definition of a standard queue.

For the second representation, we let $b=\mu^2 W_D$, combine (2.3) and (2.5) to obtain:

$$\begin{aligned} Ew' &= bE\left\{X^2 - f^2[-(w + X)]\right\} \\ &= bE_w\left\{E_x^w\, \{X^2 - f^2[-(w + X)]\}\right\} \\ &= E_w \psi(w) \end{aligned} \tag{2.10}$$

where we have defined the upper bounding function

$$\begin{aligned} \psi(y) &= bE_x^y\, \{X^2 - f^2[-(y + X)]\} \\ &= b\int_x (x^2 - f^2[-(y + x)])\, dG(x) \end{aligned} \tag{2.11}$$

Observe that due to the monotonicity of f(y), both g(y) and ψ(y) are monotone non-decreasing functions of y. Furthermore, noting that

$$w' = f(w + X) \geq f(X) \qquad \text{a.s.} \tag{2.12}$$

yields

$$E\psi(w') = E\psi(w) \geq E\psi(f(X))$$

or

$$Ew' = E\psi(w) \geq E\psi(f(X)) \tag{2.13}$$

This inequality provides a new lower bound for the stationary mean wait. We will call it the expectation lower bound and observe that it dominates the Kingman lower bound. To see this, note that (2.7) can be written as

$$Ew \geq bEf^2(X) = b[EX^2 - Ef^2(-X)] = \psi(0) \tag{2.14}$$

where we have used the relationship given in (2.4). From the monotonicity of ψ(y), the desired result follows:

$$Ew \geq E\psi(f(X)) = \int_{-\infty}^{+\infty} \psi(f(y))dG(y) \geq \psi(0) \int_{-\infty}^{+\infty} dG(y) = \psi(0) \tag{2.15}$$

In the sequel, this expectation lower bound will be compared to the Marshall bound, which is one of two bounds to be derived using the functions g and ψ.

THE INTERSECTION LOWER AND UPPER BOUNDS

In this section we will utilize the bounding functions, i.e., conditional expectations:

$$g(y) = \int_X f(y+x)\, dG(x) \tag{3.1}$$

$$\psi(y) = b\int_X (x^2 - f^2[-(y+x)])\, dG(x) \tag{3.2}$$

to obtain two companion bounds. One, the intersection lower bound, was first obtained by Marshall using a less direct proof. The second, the intersection upper bound is, to our knowledge, new. The procedure is to obtain bounds on g and ψ from derived properties of these functions. Since the stationary mean waiting time has been shown to satisfy

$$Ew = Eg(w) = E\psi(w), \tag{3.3}$$

bounds on the right-hand expectations in (3.3) will lead to bounds on the mean wait. We begin with two lemmas which give the properties of g and ψ. Because of space limitations, only the proof of the theorem will be given. (Detailed proofs can be found in [3]).

Lemma 1: $g(y)$, $y \in [0,\infty)$, is a positive, continuous, monotone non-decreasing, convex function.

Lemma 2: $\psi(y)$, $y \in [0,\infty)$, is a positive, continuous, monotone non-decreasing, bounded, concave function.

A sketch of typical bounding functions is given in Figure 1. In the figure, we show a unique intersection with the straight line $y = \ell(y)$ for each of the bounding functions. Indeed, these points y_l and y_u will provide the tightest bounds from a family of bounds to be given below.

Lemma 3: Under the assumptions $EX < 0$, $EX^2 < \infty$, and $G(0) = \Pr\{X \le 0\} < 1$, there exist unique points y_l and y_u which satisfy $y_l = g(y_l)$ and $y_u = \psi(y_u)$.

Theorem. Given a stable ($\rho<1$) standard GI/G/1 queue. The stationary mean waiting time is bounded by

$$\frac{g(y) - yg'(y)}{1 - g'(y)} \le y_l \le EW \le Y_u \le \frac{\psi(z) - z\psi'(z)}{1 - \psi'(z)} \tag{3.4}$$

where $y \ge 0$, $z \ge y_l$, y_l (Marshall's lower bound) and y_u are specified in Lemma 3.

Proof. From the definition of g, we have the relationship between successive waiting times

$$E(w_{n+1}) = E(g(w_n)). \tag{3.5}$$

Using the convexity of g, for any w_n and $\forall y \ge 0$,

$$g(w_n) \ge g(y) + g'(y)\,(w_n - y), \qquad \text{a.s.} \tag{3.6}$$

Taking expectations, substituting for $Eg(w_n)$, noting that under the stability assumption we have $\lim_{n\to\infty} Ew_{n+1} = \lim_{n\to\infty} Ew_n = Ew$, and rearranging the results yields the bound:

$$Ew \geq \frac{g(y) - yg'(y)}{1 - g'(y)} \tag{3.7}$$

With $y = y_1 = g(y_1)$, we have $Ew \geq y_1$. To show that y_1 maximizes the lower bound, we substitute $g'(y) = 1-G(-y)$:

$$Ew \geq \frac{g(y) - y + y\,G(-y)}{G(-y)} \tag{3.8}$$

The derivative of the right-hand side is $\frac{dG(-y)}{dy}\,(y - g(y))$. From this expression and the properties of g we have a unique maximum at $y = y_1$.

The proof of the upper bound proceeds in an analoqous manner. (See [3] for the details.)

EXAMPLES AND DISCUSSION

To recapitulate, we have derived two new bounds ($E\psi(f(x))$ - the expectation lower bound; $y_u = \psi(y_u)$ - the intersection upper bound) and have provided an alternate proof of the Marshall lower bound ($y_1 = g(y_1)$ - the intersection lower bound). The relationship between these and Kingman's bounds are:

$$0 \leq \psi(0) = bEf^2(X) \leq E\psi(f(X)) \leq Ew_\infty \leq y_u = \psi(y_u)$$

and

$$0 \leq y_\ell = g(y_\ell) \leq Ew_\infty \leq b\,\mathrm{Var}(X)$$

Both these orderings are dominated by $\psi(\infty)$ which is assumed finite. $\psi(0) = bEf^2(X)$ is Kingman's lower bound which is dominated by the new expectation lower bound, $E\psi(f(X))$.b Var(X) is Kingman's upper bound, and, as the examples will show, there is no order relationship between bVar(X) and the new intersection upper bound; nor is there an ordering between Marshall's lower bound y_1 and the expectation lower bound $E\psi(f(X))$. Of particular interest, then, is a comparison of these pairs of upper and lower bounds.

An initial indication of the relative merits of these bounds may be easily obtained by considering the case M/M/1, for which they and the actual mean are all easily calculable as functions of the traffic intensity $\rho = \lambda/\mu = E(S)/E(\tau)$.

For: $\rho = 1/4 \quad \psi(0) < E\psi(f(X)) < y_\ell < Ew_\infty < y_u < b\mathrm{Var}(X) < \psi(\infty)$

$\rho = 1/2 \quad \psi(0) < y_\ell < E\psi(f(X)) < Ew_\infty < y_u < b\mathrm{Var}(X) < \psi(\infty)$

$\rho = 3/4 \quad y_\ell < \psi(0) < E\psi(f(X)) < Ew_\infty < b\mathrm{Var}(X) < y_u < \psi(\infty)$

Hence, of the two relevent upper bounds, the Kingman bound $W_k = b\mathrm{Var}(X)$ and the intersection upper bound $\psi(y_u) = y_u$, neither is uniformly better in the sense that the Kingman bound dominates (is a worse upper bound) for low-to-moderate values of the traffic intensity, but becomes a better approximation at high intensities, as expected. Considering the actual numerical values (Fig. 2), however, it is seen that at high intensities, the intersection upper bound, while greater than the Kingman bound, will still be a close approximation to it.

This is due to the fact that as $\rho \to 1$ (i.e. $EX \to 0$), $W_K \to \psi(\infty)$ and since $W_K < \psi(y_u) \leq \psi(\infty)$ in this region, the intersection upper bound cannot differ significantly from W_K. While this indicates that $\psi(y_u)$ is overall a better approximation to the mean, the Kingman bound still has the advantage of being more easily calculable.

The two relevent lower bounds, the expectation bound $E\psi(f(X))$ and the interesection lower bound y_ℓ, behave in a similar manner. The expectation bound will dominate (is a better lower bound) for moderate to high intensities while the intersection lower bound will dominate for low intensities. An argument similar to the one above and justified by the examples establishes, however, that $E\psi(f(X))$ is overall a better approximation than y_ℓ. It is obviously better for moderate to high intensities where it dominates; and for low intensities it is "trapped" between $\psi(0)$ and y_ℓ while both are tending to zero as ρ tends to zero. Since we are here considering small values, the percentage difference between the bounds might conceivably be large, unlike the high intensity situation where the percentage difference between the upper bounds will not be significant. However, in the calculated examples this is not the case, as the two bounds remain remarkably close in this region. In addition, neither bound is demonstrably easier to calculate.

The same types of general behavior are reflected in all the other examples which have been investigated. It should be noted, however, that the proximity of the several bounds to the actual mean as a function of traffic intensity is very sensitive to the form of the distributions being considered. For example, with $\rho = 1/2$, the expectation lower bound is considerably closer to the mean in the D/M/1 case than in the M/M/1 case. Results of three examples (M/M/1, D/M/1, E_2/M/1) are presented in Figures 2-4.

REFERENCES

[1]. Loynes, R.M., "The Stability of a Queue with Non-Independent Inter-Arrival and Service Times", Proc. Camb. Phil. Soc., Vol. 58, pp. 497-520, 1961.

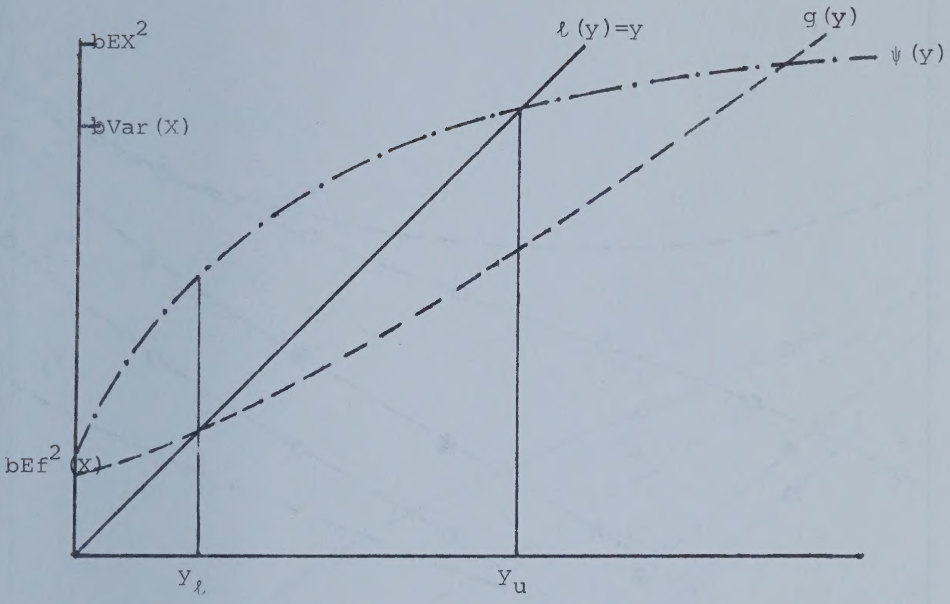
[2] Kingman, J.F.C., "Inequalities in the Theory of Queues", J. Roy. Stat. Soc. B, Vol. 32, pp. 102-110, 1970.

[3] Calo, S., Schwartz, S., "Bounds and Bounding Functions for a Stationary GI/G/1 Queue", Submitted for publication.

ACKNOWLEDGEMENTS

The research reported here was supported by the RCA Doctoral Fellowship Program and by the National Science Foundation under Grant ENG-75-09610.

FIGURE 1: Typical Bounding Functions (Moderate Traffic Intensity)



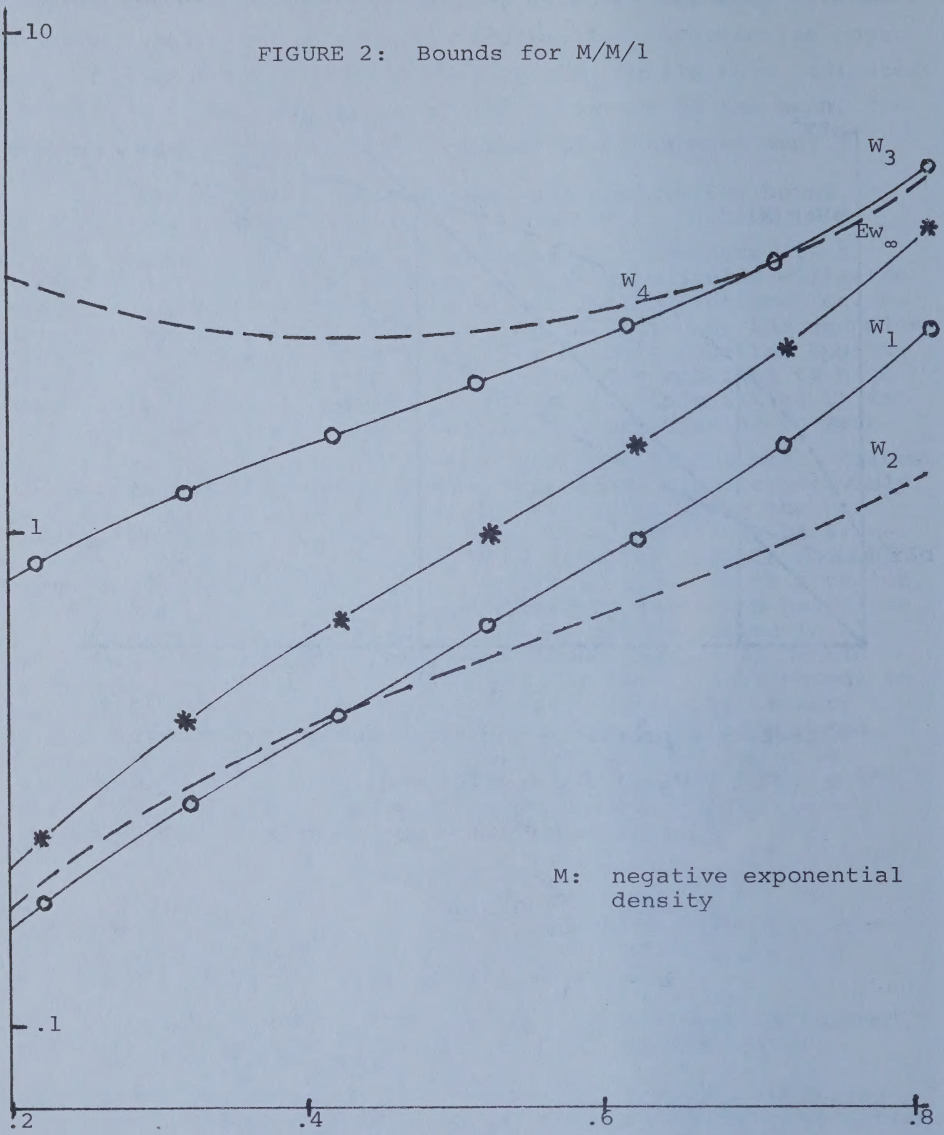
Alternate expressions for the bounding functions:

$$g(y) = y - E(-x) + C(-y)$$

$$\psi(y) = bEf^2(X) + 2b \int_{-y}^{0} C(x)dx$$

$$C(x) = \int_{-\infty}^{x} G(z)dz$$

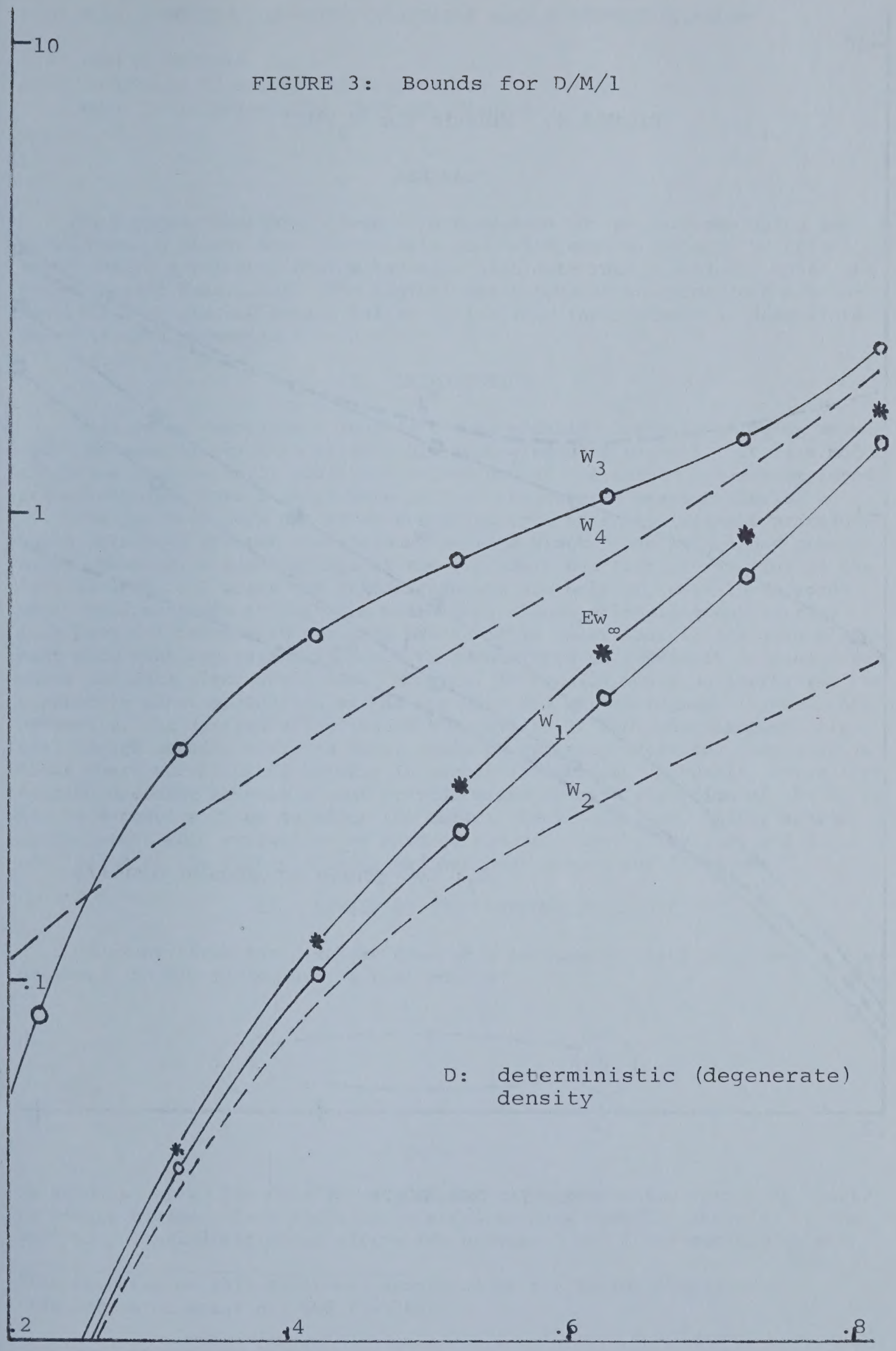
$$b = \mu^2 W_D = 1/(2\ E(-X))$$



TRAFFIC INTENSITY "ρ"

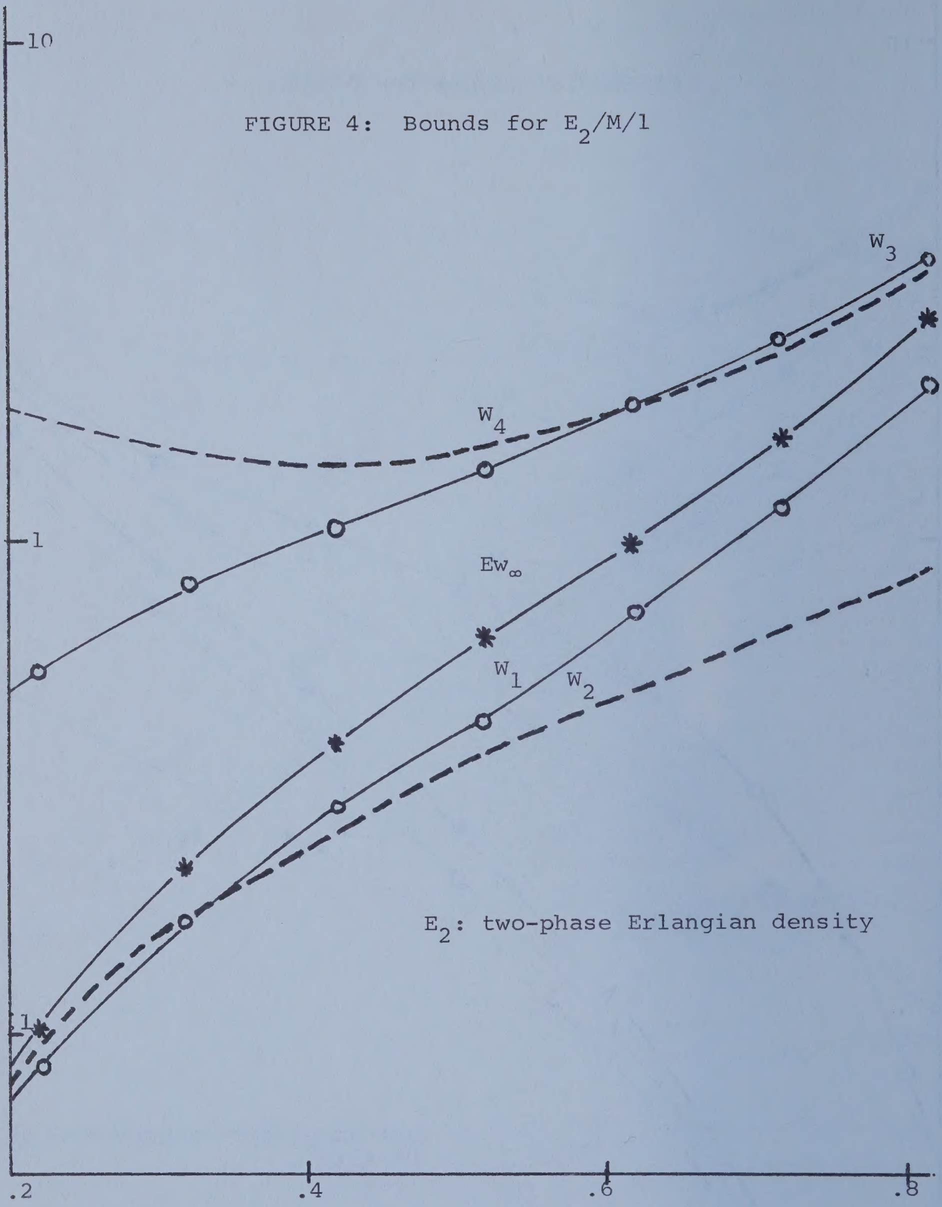
W_1 - expectation lower bound
W_2 - intersection lower bound
W_3 - intersection upper bound
W_4 - Kingman upper bound
Ew_∞ - true equilibrium mean

FIGURE 3: Bounds for D/M/1



D: deterministic (degenerate) density

FIGURE 4: Bounds for $E_2/M/1$



E_2: two-phase Erlangian density

TRAFFIC INTENSITY "ρ"

LOOP DATA COMMUNICATION WITH EXTRACTABLE ACKNOWLEDGMENT SIGNALS*

JOHN J. METZNER
Department of Electrical Engineering
Wayne State University, Detroit, Michigan

ABSTRACT

This paper describes a loop data communication network employing encoded message blocks with extractable acknowledgment signals. The object is to design a reliable system having a high data rate efficiency under adverse channel conditions. The logical operations of the procedure are described. Performance taking buffer limitations into account is determined under certain assumptions.

I. INTRODUCTION

This paper describes a loop data communication network employing encoded message blocks with extractable acknowledgment signals. It is a modification of a recently published technique[1] of the author's which employed acknowledgment signals which were part of the encoded message blocks.

The previous work described a special type of acknowledgment procedure which permitted greater transfers of message blocks than with other procedures. However, a disadvantage of the procedure was that if one link in the loop is weak, all words and acknowledgments are held up, even those words which need not pass through the weak link. Since acknowledgment carries much less information than a data block, it is clear that if the acknowledgment were sent separately it could be transmitted successfully in many cases where the data block would not. However, it usually would be inefficient to separately error-protection encode the data and acknowledgment signal. Alternately, the desired effect could be achieved if the acknowledgment signal, though encoded with the data, could be reliably extracted (decoded) in cases where the block containing it cannot be decoded. Normally, error correction encoding schemes do not provide means whereby a portion of the data can be decoded without decoding the entire block. However, coding schemes which permit this extraction of part of the data can be devised, and a recent paper of the author's described one such scheme for doing this[2].

II. REVIEW OF THE PREVIOUS PROCEDURE[1]

Consider first the simplest case of a two-station loop as shown in figure one. To fix ideas, assume that only A is sending data. The data is encoded for error protection into long blocks of binary digits. Each block is referred to as a "word". Included in the word is a label digit which alternates between 0 and 1 for successive new

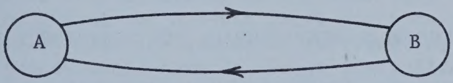


Figure 1

---

*The research on this paper was sponsored by the National Science Foundation on Grant no. ENG 75-07839

blocks. This label digit permits the receiver to tell, without reference to data content, whether a decoded block represents a new word, or is simply a repeat of a previously decoded word. Since B is not sending any words, the return signal to A can consist of an error-protected bit identifying the label of the data block most recently accepted. (This has been shown to be more efficient than sending back a digit indicating whether or not the most recent block has been decoded)[3].

Suppose now that station B also has data to send to A. Assume that word blocks in the two directions have been standardized to have equal duration and that the stations alternately send a word to each other. (This would require waiting for a return signal before sending the next word, but there are modifications available which allow continuous transmissions)[4]. In reference 1, the procedure would be for B also to use a label digit embedded in an outgoing block, and to follow the rule, "don't send a new word unless a new word is received". The procedure is illustrated in figure 2. In the figure, time increases to the right. There is a transmission waiting time gap (as between blocks a_1 and b_o at station A) as mentioned above. It must be reemphasized that this need not be wasted time; additional time slots (which may be treated as separate channels) may be interlaced to fill this gap. However, to simplify the exposition, the entire discussion of this paper deals with a single time slot.

The labels are assumed initially to have the reference 0. One station, assumed to be station A, acts as master, or initiator of new labels.

Station A sends a_1, b_o, c_1, d_o ......; Station B sends p_1, q_o, r_1, s_o .....

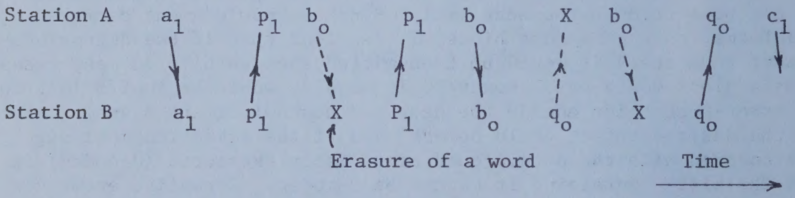

Station A
a_1 p_1 b_o p_1 b_o X b_o q_o c_1
Station B
a_1 p_1 X P_1 b_o q_o X q_o
Erasure of a word
Time

Figure 2.

When station B receives a_1, this has a label different from the reference label o, so B sends its first word block, p_1. When A receives p_1, this is a new word, and confirms a_1, so that A can send the new word block, b_o. This is not decoded by B, so that B is required to repeat p_1. If a station should happen to have no word to send, it simply sends a dummy word with the proper label. This procedure can be extended to a loop of n stations (n any integer ≥ 2), as has been described in reference 1. Only one label position is needed, regardless of the number of stations.

III. SEPARATE ACKNOWLEDGMENT SIGNALS FOR EFFICIENCY IMPROVEMENT

In the above procedure, inefficiency occurs when one or more of the links suffer a temporary reduction in capacity. For example, consider again the two-station case of figure one, where word blocks are being sent in both directions. Suppose the B→A link should become weak for a period of time, with the result that A is not able to decode any word blocks. Then, new words can not be accepted by either A or B during this time, even though the A→B link may be quite clear. If the acknowledgment signal is separated from the data or extractable, however, the situation can be greatly improved. The information rate of the acknowledgment signal is much less (of the order of one percent) than that of the data. Thus, in many cases the acknowledgment could get through the weak B→A link even though the data could not. Data transfer from A to B would not then be held up by the difficulties from B to A. Since block acceptances in the two directions no longer occur synchronously or even at the same average rate, two separate label digits are required, one covering acceptances in each direction. The desirability of an asynchronous procedure employing two separate label sequences also has been pointed out by Gray[5].

In the separate acknowledgment-data scheme, each sending block can be thought of as having two essentially unrelated parts as illustrated in figure three. Each word from A contains a label digit which alternates for successive new words, and the corresponding acknowledgment consists of the return by B of the label of the most recently decoded block. At the same time, B may send a word to A which has a label unrelated to the B→A acknowledgment label. Although the two parts of a sending logically are treated separately, it usually would be inefficient to error-protection encode the acknowledgment digit separately from the word. Reference 2 shows a way to encode the two parts together in such a way that the acknowledgment portion can be reliably extracted in cases where the channel becomes too weak to permit decoding of the whole block.

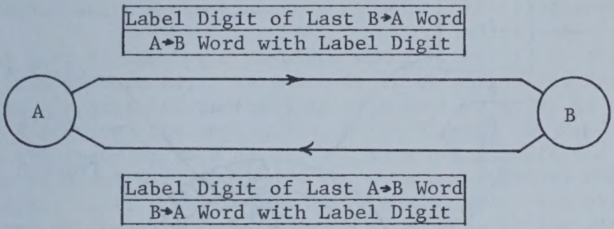

Label Digit of Last B→A Word
A→B Word with Label Digit
A
B
Label Digit of Last A→B Word
B→A Word with Label Digit

Figure 3.

IV. SEPARATE ACKNOWLEDGMENT SIGNALS IN THE N - STATION LOOP

A direct extension of the separate acknowledgment scheme to a loop of n stations requires n label digits. Other proposed systems have required only one label digit by the expedient of requiring that only one sender use a given time slot each time the signals go around the loop[6]. However, the scheme of reference 1 and its proposed modification here for separate acknowledgment signals permits from two up to n uses (depending on source and destination locations) of a time slot per circulation of the signals around the loop.

For the scheme to be practical the number of stations in the loop should not be too large. (This would probably be a practical requirement in any case). The use of n label digits is not then very inefficient. For example, if n=5, there would be a 5-bit label sequence riding on a data signal of, say, 500 bits. With the scheme of reference 2, the inclusion of the 5-bit label sequence would have an almost insignificant effect on the capacity available for data, and yet the label sequence portion could be decoded under conditions far below the threshold of decodability of the data word.

We shall assume in what follows that the word and label sequence are encoded as part of a package, but in such a way that the label sequence can be extracted in some situations where the word cannot be decoded. We shall also assume that the word never can be decoded unless the label sequence is decoded first.

The logical flow of words and label digits in a four-station loop is illustrated in figures 4 and 5. Although word and label digits are presumed to be part of a single package, it is useful to pretend the sendings are along four parallel "tracks" (refer to figure 4). Thus, on track A, station A sends the outgoing word and a label digit associated with it; on track B (analogously C,D) station A sends a label digit which is associated with B's (respectively C's,D's on other tracks) transmitted word. It must be emphasized again that these label digits are not physically sent on separate channels*; the track separation is purely for conceptual purposes. The shaded paths in figure 4 represent flow of word + label digits, while the unshaded paths represent label digits alone.

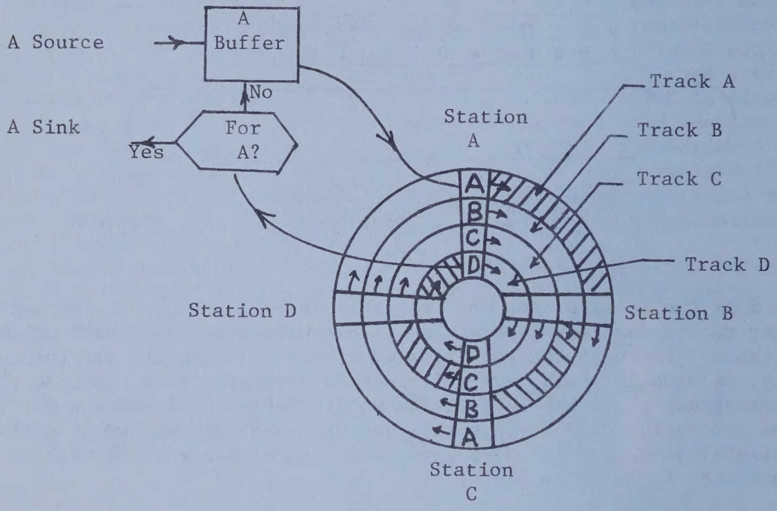


Figure 4

Each station has a buffer for storing words waiting to be sent; for clarity, only station A's buffer is shown. Words coming in to station A arrive on track D. If a new word is decoded, it is tested to see if it is for station A; otherwise, it is retained in buffer A for relaying to the destination. The buffers A,B,C, and D are assumed to be limited in capacity. The sources, however, are assumed to have their own storage capabilities for holding their messages while waiting for their acceptance by the system. In effect, the system will be designed to guarantee that virtually every message accepted by the system will be delivered error-free to its destination. We will not consider the possibility that a source may not have the storage capacity to hold its own message while waiting for acceptance. We shall only comment that a) the source data would in many cases be stored on a tape or in a computer, where storage capacity is large and b) if a source insists on generating messages at a rate uniformly greater than the system can accept them, no amount of buffering would be sufficient and loss of data is inevitable.

---

*It is possible to provide four separate channels for the four tracks; however, this is unnecessary and the need for separate error-protection encoding of each label digit would lead to inefficient use of channel space.

Figure 5 shows the flow of logic at station A with respect to tracks A and D. The simple logic for tracks B and C (shown in figure 6) is independent and can be performed in parallel. Before trying to understand figure 5, the following facts should be noted: (1) When the buffer is empty, a dummy word with the old label is sent. This dummy word is the only word which can be discarded without receiving an acknowledgment. (2) When there is only one empty space left in the buffer, this last space is reserved for relayed words only. This prevents system lockup from occurring (to be explained later). (3) In addition to the buffer, there is assumed to be a temporary location where A can examine a decoded word to determine its destination.

The first part of the logic involves examination of the track A label (if the label can be decoded). If and only if this label is decoded and is the same as the one attached to the word last sent by A, it is an acknowledgment that B received the word; thus, A can discard the word from its buffer. Next the incoming word on track D is examined. If and only if a) it is found to be a new word (based on the attached label D) and b) A is able to accept it, it is acknowledged by returning its label over track D. The reason A may not be able to accept it is that A's buffer might be full and the word might be destined for some station other than A.

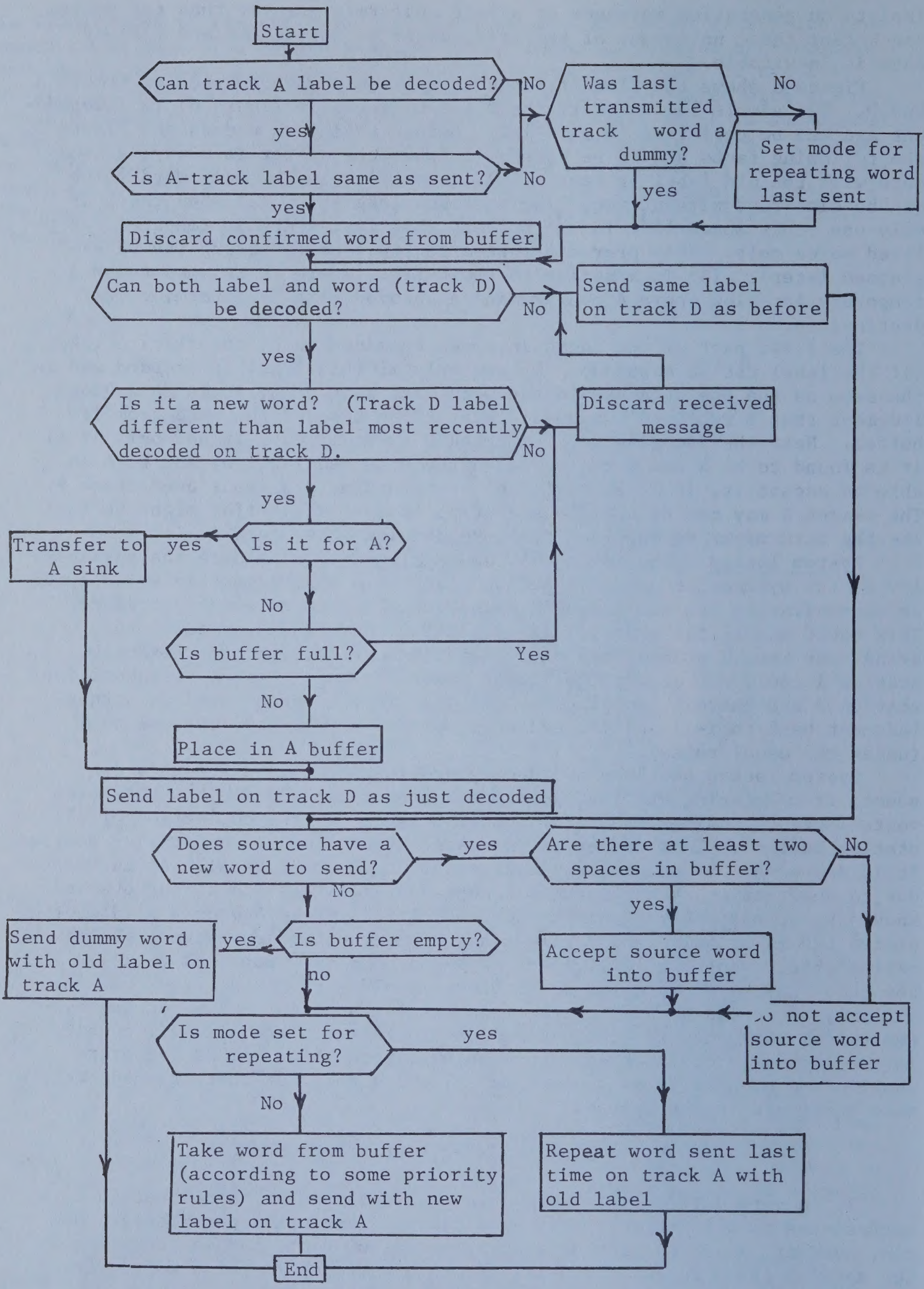
System lockup - the logic has been designed to eliminate the possibility of the system becoming "locked up", which is a situation in which further communication becomes impossible under the system rules of procedure. This could occur, for example, if all buffers were full and all words being sent were destined for other than the next station. For example, station i could not accept the incoming word from i-1, since it is not for station i and there is no place to put it. Thus i cannot send an acknowledgment back to i-1, and i-1 cannot stop sending the same message to i (under the usual rules).

System lockup has been avoided by prohibiting a word from A's own source from entering the last unfilled buffer spot. To see how this prevents lockup, consider the total number of words in the collection of all station buffers. This number is increased by new words from station sources. It is decreased by acceptances at destinations. There is also an increase due to duplication when a word is accepted at some station i, but the acknowledgment has not yet gotten back to i-1. This storage of a word in two places can exist only temporarily since the acknowledgment signal, being extractable, eventually will make its way around the loop back to i-1. If the total number of words in all buffers should rise to one less than the maximum, there can be no increase from station sources because of the system rules. The only way there can be an increase to the all-full condition is the temporary duplication phenomena explained above. This temporary duplication eventually must disappear, leaving room for another word acceptance somewhere in the system.

V. A STUDY OF NEAR FULL BUFFER EFFECTS

It is very difficult to make a general analysis of communication rate performance as a function of buffer capacities and other parameters. One can, however, make some rather specialized assumptions and use the results for special cases to gain some insight into performance for a more general range of conditions. One such special case is given here.

One concern perhaps is that the buffers may tend to stay in the full state for too much of the time, resulting in an excessive number of rejections of words to be relayed, due to lack of buffer space. To test this


Start
Can track A label be decoded?
No
yes
is A-track label same as sent?
No
yes
Was last transmitted track word a dummy?
No
yes
Set mode for repeating word last sent
Discard confirmed word from buffer
Can both label and word (track D) be decoded?
No
Send same label on track D as before
yes
Is it a new word? (Track D label different than label most recently decoded on track D.
No
Discard received message
yes
Transfer to A sink
yes
Is it for A?
No
Is buffer full?
Yes
No
Place in A buffer
Send label on track D as just decoded
Does source have a new word to send?
yes
Are there at least two spaces in buffer?
No
No
yes
Send dummy word with old label on track A
yes
Is buffer empty?
no
Accept source word into buffer
Do not accept source word into buffer
Is mode set for repeating?
yes
No
Take word from buffer (according to some priority rules) and send with new label on track A
Repeat word sent last time on track A with old label
End

Track A & D Logic

Figure 5.

effect, assume a model where each source has data available to send at all times, so that whenever the buffer is free to accept a source word there is one for it to take in. This assumption should yield a good test of near-full buffer performance, though it does not reveal underutilization phenomena due to occasionally empty buffers.

Focussing attention on one station in the loop, we shall assume that of the words coming into the station, a portion α require to be relayed, while a proportion $1-\alpha$ are destined for the given station. To justify a fixed probability α that the incoming word is required to be relayed, one could make a somewhat artificial assumption that the other station exercises some choice as to which words to take in from its source and which words to send out in order to keep the appearance of a fixed probability α. Let q_{in} (p_{in}) be the probability that an incoming word can (cannot) be decoded, assumed independent from word-to-word. Similarly, let q_{out} (p_{out}) be the probability that an outgoing word will (will not) be decoded successfully and accepted by the next station. The term p_{out} includes those cases where the next station cannot accept due to a full buffer. Independence of these outgoing acceptances also will be assumed, although the proportion of rejections which are due to a full buffer at the next station are not really independent. Where this proportion is found to be small, the independence assumption should be a good approximation. We shall also assume that acknowledgment signals always get through. This is a reasonable assumption in view of their low information rate and extractability.

We have assumed that even when the buffer is full, there is an additional memory location where an incoming word can be decoded and examined. If a word is decoded but cannot be accepted because the buffer is full and it is required to be relayed, we shall assume that this decoded result is left where it is. This can be done because this word must be the word which the prior station will continue to send until acknowledged.

With the above assumptions, we can define three states with respect to a station's buffer.

FR - buffer is full and there is a word which has been decoded but not acknowledged.

F - buffer is full but there is not a decoded word which has not been acknowledged.

A - buffer is one from full.

If the buffer state is A and a word is acknowledged and can be discarded from buffer, a word is taken from the source into the buffer if and only if there is no incoming word which must be relayed. Thus, the constant availability of source words prevents the buffer from ever being two or more from full.

Let Q_{FR}, Q_F, and Q_A be the limiting probabilities of states FR, F, and A, respectively. It is readily shown from figure 7 that

$$Q_{FR} = \frac{(\alpha\, q_{in}\, p_{out})^2}{(q_{out} + \alpha\, q_{in}\, p_{out})^2 - \alpha\, q_{in}\, q_{out}} \tag{1}$$

$$Q_F = \frac{q_{out}\, q_{in}\, \alpha\, p_{out}}{(q_{out} + q_{in}\, \alpha\, p_{out}{}^2 - \alpha\, q_{in}\, q_{out}} \tag{2}$$

Let r be the fraction of incoming decodable words which must be rejected due to lack of buffer space.

$r = \alpha\, p_{out}\, Q_F + p_{out}\, Q_{FR}$ which reduces to

$$r = \frac{q_{in}\, \alpha^2\, p_{out}^2\, [q_{out} + q_{in}\, p_{out}]}{(q_{out} + q_{in}\, \alpha\, p_{out})^2 - \alpha\, q_{in}\, q_{out}} \tag{3}$$

To select an appropriate α, suppose that each source is equally likely to send a message to each of the other stations. If there are n stations, station 1 originates messages to n-1 other stations, and $\sum_{i=1}^{n-2} i = \frac{(n-1)(n-2)}{2}$ source-destination paths pass through station 1 as a relay station. Thus it is logical to set

$$\alpha = \left[\frac{(n-1)(n-2)}{2}\right] \Big/ \left[\frac{(n-1)(n-2)}{2} + n-1\right] \quad \text{or}$$

$$\alpha = \frac{n-2}{n} \tag{4}$$

Figure 8 shows plot of r versus p_{out} for various parameter values. Setting $p_{in}=0$ represents a worst case as far as buffer space rejection probability is concerned. Cases with $p_{in}=0$ and with $p_{in}=p_{out}$ are shown for $\alpha=\frac{1}{3},\frac{1}{2},$ and $\frac{2}{3}$, corresponding to 3,4, and 6 station loops, respectively. A system most likely would be operated with p_{out} less than 0.2, for which less than $5\frac{1}{2}$ percent of the decoded incoming words would be rejected due to lack of buffer space in all of the cases shown.

It is of some interest to see the number of words taken from the source relative to the number relayed. Let s be the number of words per unit time taken from the source. Then referring to figure 6, it can be shown that

$$s = \frac{q_{out}^3 (1-q_{in}\, \alpha)^2}{(q_{out} + q_{in}\, \alpha\, p_{out})^2 - \alpha\, q_{in}\, q_{out}} \tag{5}$$

Figure 9 is a plot of s versus p_{out} for both $q_{in}=q_{out}$ and $p_{in}=0$, assuming $\alpha=\frac{1}{2}$. The straight line $q_{out} = 1-p_{out}$ represents the sum of the average number of source words and number of relayed words per unit of time. For the $p_{in}=0$ case, the straight line $q_{out}-\alpha$ ($q_{out}>\alpha$) represents the average number of source words that would be sent per unit time if all words for relaying could be accepted. For $p_{out}<.2$, it can be observed that s is close to $q_{out}-\alpha$.

VI. CONCLUSIONS

Reliable loop data networks can be designed which employ extractable acknowledgment signals and which can permit from 2 up to n uses of each time slot per circulation around an n-station loop. An example presented suggests that the percentage of words rejected due to a full buffer would be small in most cases.

REFERENCES

1. J.J. Metzner, "The New-Word Policy and Decision Feedback in Loop Data Communication Networks", IEEE Transactions on Communications, vol. COM-21, no. 6, pp. 727-730, June 1973.

2. J.J. Metzner, "A Technique for Separation of Data and Acknowledgment Signals in Two-Way Feedback Communication", IEEE Transactions on Communication, vol. COM-22, no. 6, pp. 881-883, June 1974.

3. J.J. Metzner, "The New-Word Policy for Two-Way Feedback Communication", IEEE Transactions on Communications, vol. COM-20, no. 5, pp.1041-1043, October 1972.

4. J.J. Metzner and K.C. Morgan, "Word Selection Procedures for Error-Free Communication Systems", New York University, Second Scientific Report for Air Force Cambridge Research Laboratories, Contract AF19 (628)-4321, June 1965.

5. J.P. Gray, "Line Control Procedures", Proc. IEEE, vol. 60, pp.1301-1312, November 1972.

6. R.B. Anderson, J.F. Hayes and D.N. Sherman, "Simulated Performance of a Ring-Switched Data Network", IEEE Transactions on Communications, vol. COM-20, no. 3, part II, pp. 576-591, June 1972.

7. R.E. Kahn and W.R. Cronther, "Flow Control in a Resource-Sharing Computer Nerwork", IEEE Transactions on Communications, vol. COM-20, no. 3, part II, pp. 539-546, June 1972.

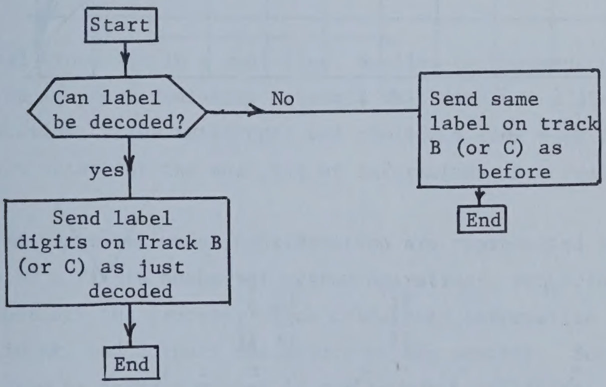

Start
Can label be decoded?
No
Send same label on track B (or C) as before
End
yes
Send label digits on Track B (or C) as just decoded
End

Figure 6.

Track B (or similarly C) Logic.

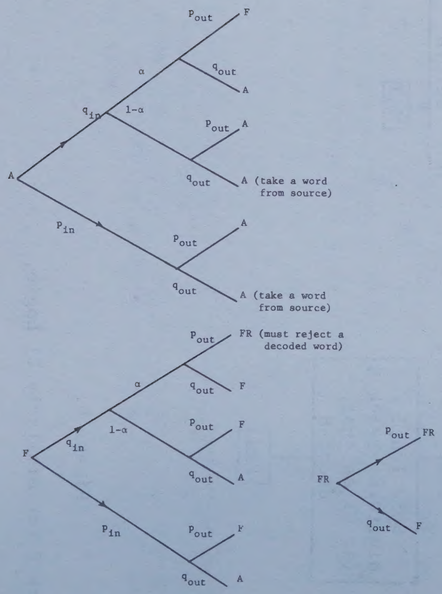

A
q_{in}
α
p_{out}
F
q_{out}
A
$1-\alpha$
p_{out}
A
q_{out}
A (take a word from source)
p_{in}
p_{out}
A
q_{out}
A (take a word from source)
F
q_{in}
α
p_{out}
FR (must reject a decoded word)
q_{out}
F
$1-\alpha$
p_{out}
F
q_{out}
A
p_{in}
p_{out}
F
q_{out}
A
FR
p_{out}
FR
q_{out}
F

Figure 7

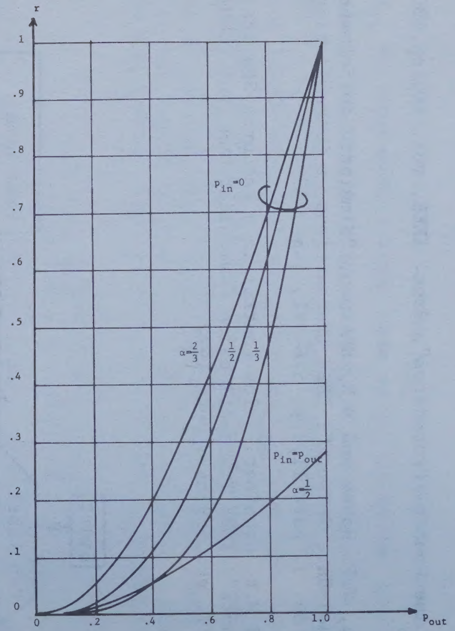


Figure 8

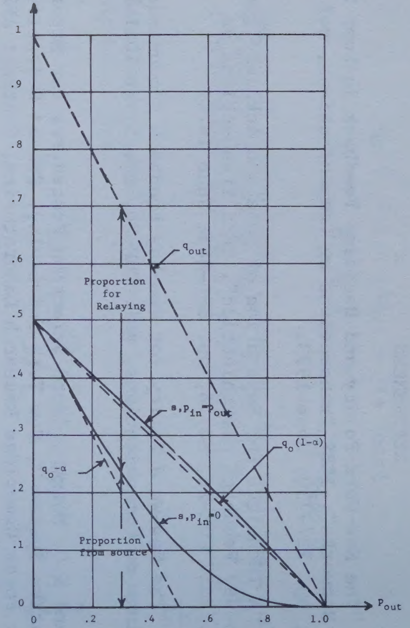


Figure 9

ANALYSIS OF REAL TIME SYSTEMS: AN INFORMATION THEORETIC APPROACH

W. R. ADRION
P. A. FRICK
P. A. SZULEWSKI
Oregon State University
Corvallis, Oregon 97331

ABSTRACT

In this paper a formal model for real-time systems is proposed. The processes under consideration are restricted to a specific class of algorithms; specifically those processes that can be implemented in an interrupt driven micro or mini processor. This model permits analysis of information flow in the computation component of the real-time system and supplies a measure of information loss in the process.

The proposed model is based on a real-time computation schema which is an interpreted directed graph. Interrupts can be represented in a real-time process. System operations are defined on the memory and the model permits an operation classification. An instantaneous discrete time step description of the system is defined. A qualitative measure of system performance is derived. The concepts of maximum information capacity, computational capacity and processing bandwidth are introduced and developed.

Some properties of the proposed model are compared with other schemata. Example algorithms are analyzed.

Computational processes in a real-time, on-line environment can be represented by the proposed real-time schemata which include a directed graph model of a computation, allow interrupts and provide a time base for comparison of computation rates for the analysis of information in a real-time system.

The real-time processes under consideration are represented by an algorithm consisting of a finite number of system operations, requiring input information to initiate the process, which calculates information as a function of the input, and outputs the result of the process. Such processes can be viewed as being composed of two separate and distinct parts, calculations on information in memory and information transfers to and from memory. Such information, transfers, in the systems under consideration, typically occur as interrupts, before, during or after a calculation.

A general schema model, developed by Y. I. Ianov [2] provides adequate

---

*This work was partially supported by the National Science Foundation under Grant GK 39914.

analysis of computations where simple sequential control is assumed. Ianov's schemata formulation was modified by R. M. Karp and R. E. Miller [3] to include parallel computations. Both of these schemas lack the ability to model the random interrupt in a computational sequence and are weak in representing the time element in any computation. The real-time schema the authors introduce can model interrupt behavior of a real-time system.

Definition 1 A real-time computation schema is a triple $S=(M,\Omega,T)$ where:

- i) M is a finite set of named memory locations $\{m_0,m_1...m_n\}$, m_0 is the null location,
- ii) Ω is a finite set of elementary operations (mappings) $\{a_1,a_2...a_k\}$. Associated with each $a_i\varepsilon\Omega$ are:
  - a) $Da_i\subseteq M$, the domain locations;
  - b) $Ra_i\subseteq M$, the range locations;
  - c) $\gamma_i\varepsilon M$, a condition flip-flop.
- iii) T is the control graph and $T = \{Q,q_0,Q_T,\Sigma,\mu,\delta\}$ where:
  - a) Q is a finite set of states $\{q_0,q_1...q_k\}$;
  - b) $q_0\varepsilon Q$ is the unique initial state;
  - c) $Q_T\subseteq Q$ is a set of acceptor states;
  - d) Σ is an alphabet corresponding to the allowed set of operations, $\Sigma = \{\sigma_i|a_i\varepsilon\Omega\}$
  - e) μ is a LIFO stack
  - f) δ is a partial transition function $\delta:Q\text{x}\mu\text{x}\Sigma\to Q\text{x}\mu$

The sequence of allowed operations, corresponding to a computation in real-time, is enforced by the control graph T. The nodes in the graph represent the state of the system's memory and transitions from one state to the next in discrete time by the directed edges between nodes. Construction of a real-time schema control graph for a given process is based on a program graph representing the algorithmic part of the computation. The graph must allow interrupt capability at each node. Interrupts will be handled on a last request priority, i.e. an interrupt may interrupt another interrupt. However, once a request is initiated on any interrupt I/O channel, all further requests to that channel will be disabled until that request is satisfied. Interrupt and interrupt returns are handled by the LIFO stack μ which is used to stack return state information.

As an example, consider a simple real-time process corresponding to an interrupt driven, one-step table lookup procedure. The system has one input (i_b) and one output channel (o_b), and a small memory $M = \{m_0,m_1...m_{10}\}$. Figure 1 shows a program graph for the example which does not employ the LIFO stack.

Note that even for this simple one-step example a program graph becomes quite complex when nested levels of interrupts are considered. The graph allows the possibility of a countably infinite number of interrupts to occur. A situation where this is possible is when a computation becomes I/O bound.

An interrupt driven program graph is not finite state. To simplify the schema control graph, the computation is viewed in two parts, a calculation subgraph representing the algorithm and an interrupt subgraph. The interrupt subgraph will have a finite number of entry points corresponding to the I/O channels defined and one common exit, the return state, which uses the return state LIFO stack to provide next state information. To reconstruct the complete program graph, the interrupt subgraph is used to augment each node of the calculation subgraph to the necessary levels. Figure 2 is such a partition on the graph given in Figure 1.

In order to assign "meaning" to the evaluation of a calculation within a particular schema, a precise interpretation must be specified. The idea of an interpreted schema is similar to that of Karp and Miller [3] with the addition of the notion of time.

Definition 2 An interpretation I of schema S with respect to time is specified by

i) An initial time t_0, and discrete ordered, but not necessarily uniform time steps $t_1, t_2, \ldots t_k, \ldots$

ii) A function C which associates with each element $m_i \varepsilon M$, a set $C(m_i)$, the possible contents of m_i. The notation $[m_i(t_j)]$ represents a specific content of location m_i at time t_j, $[m_i(t_j)] \varepsilon\ C(m_i)$ and $[M(t_j)]$ the total contents of memory at time t_j,

$$[M(t_j)] \quad \varepsilon \underset{m_i \varepsilon M}{X} \quad C(m_i).$$

iii) An initial contents of memory $[M(t_0)]$

iv) For each $a_i \varepsilon \Omega$

a) the time to execute operation a_i, $\tau(a_i)$

b) a function
$$F_{a_i} : \underset{m_j \varepsilon D_{a_i}}{X} C(m_j) \rightarrow \underset{m_j \varepsilon R_{a_i}}{X} C(m_j)$$

v) A function
$$G: \underset{m_i \varepsilon M}{X} C(m_i) \rightarrow \underset{\gamma_i \varepsilon M}{X} C(\gamma_i)$$

An interpretation of the operations for the example of Figure 2 is given in Table 1.

Table 1. An Interpretation for the Operations of Figure 1.1.

Operation	Type	D_{a_i}	R_{a_i}	Operation Action
a_1	Conditional Jump	---	---	State Transition
a_2	Conditional Jump/Table Lookup	$m_2..m_9$	m_1,m_o	$m_{[m_o]} \to m_1, 0 \to m_0$
a_3	Unconditional Jump	---	---	State Transition
a_4	Vectored Interrupt	---	---	$\downarrow\mu$
a_5	Input	i_b	m_0	$[i_b] \to m_0$
a_6	Vectored Interrupt	---	---	$\downarrow\mu$
a_7	Output	m_1	o_b	$[m_1] \to 0_b$
a_8	Return	---	---	$\uparrow\mu$

Note: The notation $\downarrow\mu$ and $\uparrow\mu$ represents pushing and popping return states on stack μ respectively.

The condition flip-flops $\gamma_1, \gamma_2, \ldots \gamma_8$ are set as follows

$$\gamma_1 = \begin{cases} 1, [M_0] = 0 \\ 0, \text{Otherwise} \end{cases}$$

$$\gamma_2 = \begin{cases} 0, [M_0] = 0 \\ 1, \text{Otherwise} \end{cases}$$

$$\gamma_3 = \gamma_4 = \gamma_5 = \gamma_6 = \gamma_7 = \gamma_8 = 1$$

Furthermore, the set of acceptor states $Q_T = \{q_1\}$

A real-time computation corresponds to a well defined sequence of operations, which include interrupt operations, that begins in the initial state and either iterates through or terminates in some designated acceptor states. An acceptor state in the graph model, represents a logical point in the computation where the calculation on input information is in an acceptable state for output. This may correspond to a terminal state in a fixed length algorithm or an appropriate sample state in an inerative algorithm. Acceptability is a function of the "useful information" calculated in a process.

Definition 3 A string $\sigma \varepsilon \Sigma^*$ is well defined for a schema S if for all prefixes β of σ, δ (q_0, μ, β) is defined on S, where $\mu(t_0) = \emptyset$

If $\ell(\sigma)$ is the length of the string σ, eg. if $\sigma = a_{i1}, a_{i2} \ldots a_{ik}$, $\ell(\sigma) = k$, then the following may be stated.

Definition 4 The set of k-bounded well defined sequences for a schema S is a finite set $\Sigma^K = \{\sigma\varepsilon\Sigma^* | \ell(\sigma)\underline{<}k \text{ and, } \sigma \text{ is well defined}\}$

For any well defined sequence $\sigma_i = a_{i1}, a_{i2} \ldots a_{ik}$, the time elapsed (machine time) after j operations $i\underline{<}j\underline{<}k$ is calcluated by the formula

$$t_j = [\sum_{\ell=1}^{j} \tau(a_{i\ell})] + t_0 \quad j=1\ldots k, \text{ and the total elapsed time } \tau(\sigma_i) = t_k$$

Definition 5 Let $\sigma_i = a_{i1} \ldots a_{ik}$ be a k-length sequence in Σ^K for a schema S. The ordered set of states $\{q_i\} = \{q_0, q_{i1} \ldots q_{ik}\}$, of length k+1 is the q-set associated with the computation σ_i, where q_0 is the initial state and for $i\underline{<}j\underline{<}k$, $q_{ij}\varepsilon Q$, $a_{ij}\varepsilon\sigma_i$ and $\delta(q_{ij-1}, \mu, a_i) = (q_{ij}, \mu')$

Definition 6 A useful computation occurs in a sequence $\sigma_i\varepsilon\Sigma^K$ under schema S if $\{q_i\}$, the associated q-set, contains at least one element $q_i\varepsilon Q_T$, the set of acceptor states.

The q-set concept is essential for the analysis of useful information production by a computation. Given a sequence of operations, a q-set can be generated and the number of occurrences of acceptor states in it is directly proportional to the useful information generated by the process.

In the example of Figure 1, $\sigma_0 = a_1 a_4 a_5 a_8 a_2 a_3 a_6 a_7 a_8$ is a well defined sequence. The associated q-set is $\{q_0\} = \{q_0 q_0 q_2 q_3 q_0 q_1 q_0 q_4 q_3 q_0\}$ and the computation is useful when state q_2 is reached.

Earlier definitions introduce the concept of length bounded strings. In order to develop a notion of time bounded strings and information production rates, the following are defined.

Definition 7 The set of well defined sequences bounded by time t is defined.

$\Sigma_t = \{\sigma\varepsilon\Gamma | \tau(\sigma)\underline{<}t\}$

Where Γ is the set of all well defined strings $\Gamma = \underset{i}{U} \Sigma^i$.

Definition 8 A trace on Σ_t is a set of q-sets $T_t = \{\{q_i\} | \sigma_i\varepsilon\Sigma_t\}$

Definition 9 An incremental trace over time interval

$T_{t,\underset{\downarrow}{\hat{\tau}}} = \{\{\upsilon\} | \{q_i\}\varepsilon T_{t-\hat{\tau}}, \{q_i, \upsilon\}\varepsilon T_t\}$

where $\{q_i, \upsilon\}$ is a q-set which consists of the ordered set $\{q_i\}$ followed by the ordered set of states $\{\upsilon\}$

Examination of the elements of $T_{t,\hat{\tau}}$ will indicate generation of useful information during the time interval $(t-\tau, t)$. As was suggested above, useful information depends on the occurence of acceptor states in a particular $\{\upsilon\}$. Mere occurence of an acceptor state does not in itself guarantee the generation of useful information, but rather information was calculated.

Whether this calculation was performed on valid input, or was a repitition or was ignored as an output are also criteria. Consider

$$F(t,\hat{\tau})=\text{Prob}\{\{\upsilon\},\ R(\upsilon)\}$$

where $\{\upsilon\}$ is an element of $T_{t,\hat{\tau}}$ for some t and $\hat{\tau}$ and $R(\upsilon)$ is a rule for determining when useful information is generated from υ. The probability distribution $F(t,\hat{\tau})$ characterizes the useful information produced in the interval $(t-\hat{\tau},t)$.

The rule R results from the mathematical realization of input/output data flow. Depending on the process and algorithm used a miriad of possible models exist to represent the flow.

One such realization would be to consider the arrival of input and output data as independent Poisson processes x_t and y_t and depict the information flow (input/output relationship) as

$$Z_t=\text{Max}(0,X_t-Y_t)$$

For purposes of illustration, assume that Z_t is in fact a positive parameter compound Poisson process. It is sufficient to allow jumps in Z_t only of magnitude +1 (corresponding to an input) and -1 (corresponding to an output). Assume further that Z_t with rate λ experiences jumps at $t_1,t_2,\ldots$ with magnitude expressed by the random variables $x_1,x_2,\ldots$ which are independent and identically distributed, i.e. $P(x_i=1)=p$.

In an interval $(0,t]$ let N^1 and N^{-1} be the number of jumps of magnitude 1 and -1 respectively. It is well known that N_t^1 and N_t^{-1} are now independent Poisson processes with rates

$$\lambda^1=\lambda P\{x_i=1\}=p\lambda \text{ and } \lambda^{-1}=\lambda P\{x_i=-1\}=(1-p)\lambda$$

and

$$Z_t = N_t^1 - N_t^{-1}$$

Since $E[x_i] = 2p-1$, if N_t represents the process corresponding to the total number of jumps in Z_t on $(0,t]$ i.e.

$$N_t=N_t^1+N_t^{-1} \quad (N_t\ t\geq 0 \text{ in itself being Poisson})$$

then

$$E[Z_t|N_t=n] = (2p-1)N_t$$

and $E[Z_t] = (2p-1)E[N_t] = (2p-1)\lambda t$

since $E[N_t] = E[N_t^1+N_t^{-1}] = p\lambda t + (1-p)\lambda t = \lambda t$

Both N_t^1; N_t^{-1} and Z_t being Poisson, each have independent interarrival times with exponential distribution. The rule $R(\upsilon)$ is therefore always expressible in terms of either the arrival or interarrival times or both.

For the example of Figure 1, a possible rule R would require t to be the time of an output interrupt and $\hat{\tau}$ the time elapsed since the last prior input interrupt. The decision would then be based on whether υ contains an acceptor state. The probability distribution $F(t,\hat{\tau})$ for this rule, assuming $\tau(a_i)=\tau_0$ for all a_i, is given by Figure 3.

CONCLUSION

A framework has been suggested and applied to a simple table lookup procedure. Any measure on useful information in a real-time system would depend on the input-output relationship Z_t and the rule applied. Multi-step, non-iterative algorithms behave similar to a table lookup procedure. For iterative algorithms a different rule must be applied.

REFERENCES

1. W. Adrion, R. Rink and S. Sachs, "Computational Capacity in Digital Systems," *Proceedings of the 1975 ACM Computer Science Conference*, February, 1975.
2. Y. I. Ianov, "The Logical Schemes of Algorithms," *Problems of Cybernetics*, Vol. I, pp 75-127, 1958.
3. R. Karp and R. Miller, "Parallel Program Schemata," *Journal of Computer and System Sciences*, Vol. 3, 1969.
4. J. Rutledge, "On Ianov's Program Schemata," *Journal of the Association for Computing Machinery*, Vol. 11, 1964.

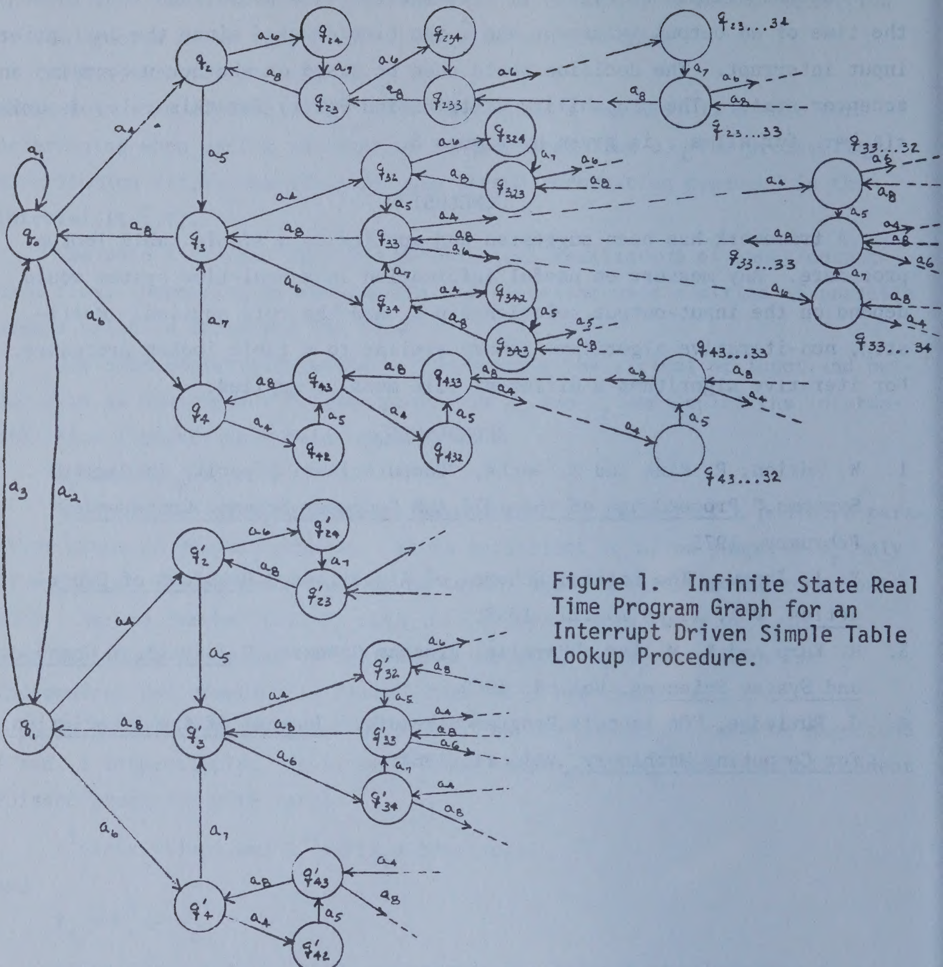

q₀
q₁
q₂
q₃
q₄
q₂₄
q₂₃
q₂₃₄
q₂₃₃
q₂₃...₃₄
q₂₃...₃₃
q₃₂
q₃₃
q₃₄
q₃₂₄
q₃₂₃
q₃₄₂
q₃₄₃
q₃₃...₃₂
q₃₃...₃₃
q₃₃...₃₄
q₄₂
q₄₃
q₄₃₂
q₄₃₃
q₄₃...₃₃
q₄₃...₃₂
q′₂
q′₃
q′₄
q′₂₄
q′₂₃
q′₃₂
q′₃₃
q′₃₄
q′₄₂
q′₄₃
a₁
a₂
a₃
a₄
a₅
a₆
a₇
a₈

Figure 1. Infinite State Real Time Program Graph for an Interrupt Driven Simple Table Lookup Procedure.

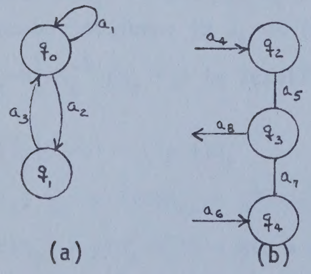

q₀
q₁
a₁
a₂
a₃
q₂
q₃
q₄
a₄
a₅
a₆
a₇
a₈
(a)
(b)

Figure 2. A Calculation Graph and Its Associated Interrupt Subgraph.

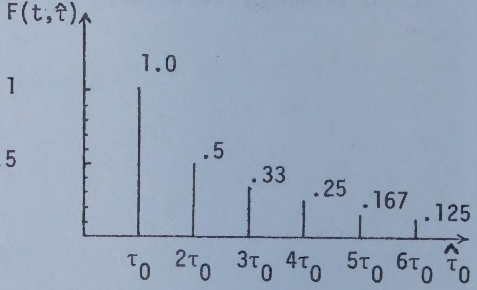


Figure 3. Useful Information Production.

FAIL RESISTANT MULTIPROCESSING NETWORKS

GERARD G. L. MEYER
The Johns Hopkins University
Baltimore, Maryland 21218

ABSTRACT

This paper discusses a new criterion for the design of fail resistant networks of computing elements. Two examples of such networks are presented. The first network is designed for the solution of filtering problems and the second network is specialized to the solution of optimal control problems with state constraints.

INTRODUCTION

Given a particular organization of computing elements, "efficient" methods may be synthesized for a given problem P to exploit the organization's structure. Conversely, given P, computing elements may be organized and segmented methods designed to "efficiently" use the problem's structure. The design criterion in both cases usually implies some sort of minimization of an execution time or space-time measure of the process. This criterion does not take into account the importance of maintaining the process integrity when one or more computing elements fail. Existing approaches used to avoid this difficulty emphasize redundancy in the system design and implementation of recovery procedures. In these instances the criterion is that the remaining network still solves the original problem P in the presence of a class C_f of hardware faults. Achievement of this objective frequently necessitates a substantial increase in complexity and cost. The criterion proposed could be considered as a dual to the preceding one. Instead of specifying a class C_f of failures, a class C_p of admissible problems P_a is specified. Given C_p the design objective is to synthesize a computing network so that in the absence of faults the original problem P is solved and in the presence of faults, the approximating problems solved by the remaining part of the network are in C_p. The specification of C_p imposes the basic structure for the segmentation of the algorithm and the organization of the computing elements so as to realize the desired fail resistant properties of the process. The proposed design criterion yields parallel processing networks of computing elements which exhibit highly desirable characteristics: (i) in the absence of faults P is solved using the full parallel capabilities of the networks; (ii) in the presence of faults, admissible problems in C_p are solved. This last property is essential in that it allows maintenance of the networks without stopping the production of results. The use of the proposed criterion will be illustrated by the presentation of networks of processing elements for optimal control and filtering.

NETWORK CONSTRUCTION SPECIFICATIONS

The introduction of inexpensive computers make it possible to consider networks of computing elements for solving problems. The design of such networks depends on the types of tasks for which the networks are to be used. To clearly state our objectives, we begin by discussing the specifications of networks according to types of tasks and modes of operation.

Types of Tasks

We chose to classify the types of tasks which the networks must perform on the basis of the number of different problems which must be solved and whether or not a description of these problems is known in advance. The following categories are typical:

1. Many different problems whose descriptions are not known in advance.
2. Many different problems whose descriptions are known in advance.
3. One problem whose description is known in advance.

The first category is the most general and is the situation at many centralized computing facilities which must serve a great variety of users. An example is the computing center of a large university. The second category is easier to deal with. Since the problems are known in advance the efficiency with which a particular computer structure can deal with them may be predicted. This type of task is ideally suited to organizable networks of computing elements. We will be most concerned with the last category since after having treated it; we also have an approach to dealing with the second category. The third category includes situations where a single problem is repeatedly solved as in the control of a continuous process. As an example one might imagine the problem of controlling distances between subway trains.

Modes of Operation

A network of computing elements may be used off-line or on-line. The construction criteria in either cases are different.

Off-line mode: Space-time considerations are usually important for economic reasons. Time constraints for solving problems off-line are relaxed over time constraints for solving problems on-line. It is important though to detect faults early so that users develop a high confidence in the accuracy of the system performance. Once a fault is detected, processing may stop and another computer may be used.

On-line mode: On-line mode generally associates a minimum running time with a problem to be solved. One aspect of parallelism is to decrease computation time. It is at least as important to detect faults in on-line systems as in off-line systems. Parallelism should reduce the variety of tests required to detect faults. Once a fault is detected, immediate action is required to protect the integrity of the computations. Parallelism in the form of redundant hardware is often used to minimize the time required to resume normal system operation.

In this paper we are only interested in the on-line processing of tasks of type 3. In other words, we have to solve repeatedly on-line one problem which is known in advance.

The on-line detection of faults and the on-line actions to be taken in the presence of faults influence also the design of computing networks.

Fault Detection and Corrective Actions

The problem of faults may be decomposed into two parts: (i) the detection of the faults and (ii) the action to be followed when confronted by the detected faults. These two aspects influence a network structure. Fault detection is based on the fact that the response of a device is different from what it should have been. We need to store information about input tests, test programs and the corresponding outputs. Fault detection is simplified if several similar devices are used in the network because we need not store the outputs. Note that our network is used on-line and therefore, the tests have to be performed every now and then. For example, a central clock may send a test signal every 100 ms to all the computing elements in the network at which time they all run the same tests for 1 ms. A comparing device associated with the central clock verifies that the outputs of the devices are identical content-wise and time-wise.

The second aspect, i.e., the selection of the actions to be taken in the presence of faults is as important as the detection proper for on-line application. Three approaches are possible (i) active duplication, (ii) passive duplication and (iii) degradation.

In the case of active duplication, the devices are duplicated as many times as money or good sense allows and all the devices are being used. When a fault occurs, there is no need to lose time starting devices and there are no transients. In the case of passive duplication, the unused devices are not turned on but are ready and in place. In case of faults, the necessary devices are turned on and a switching network switches off the faulty devices and switches on the new or stand-by ones. In the last case i.e., degradation, the faulty devices are turned off and the results they produce are ignored by the remaining devices. The purpose of fault detection has to do with not letting corrupted results influence the non-faulty parts of the network.

All these construction considerations lead us to believe that for solving one on-line repetitive problem, the network of computing elements should consist of identical devices organized in such a way as to allow for a graceful degradation in case of faults.

PROBLEM SPECIFICATION

The usual approach to problem specification in a faulty environment consists of defining the problem P to be solved and a class of faults C_f. The construction constraints are such that if no fault occurs P is solved and if a fault in C_f occurs then P is still solved. Our approach consists in relaxing the construction constraints in the presence of faults. In addition to the problem P and the class of faults C_f, we defined a class C_p of admissible problems which approximate P in some sense. We recognize that if a fault in C_f occurs, it may be impossible to solve P unless a redundant component is available to replace the defective one. Instead, if a fault C_f occurs we require that the network be able to solve a problem in C_p, the admissible class of approximating problems. Often this is acceptable when it is considered that it is no longer necessary to duplicate the computing hardware in a brute force attempt to protect against faults.

SEGMENTED ALGORITHMS

A segmented algorithm is composed, as indicated by the name, of segments or modules. The segments must have the property that (i) if they are all present the given problem P is solved and (ii) if some of the segments are unavailable then a problem in C_p is solved.

The modules of a segmented algorithm must be implemented on suitable hardware in a way which satisfies the design constraints. Specifically a fault in C_f must not result in the nonavailability of too many modules. This condition results in a lower bound on the number of computing elements which must be used in the network. For example, suppose that a segmented algorithm consists of N segments and assume that the disappearance of more than two segments results in a problem outside C_p being solved. Furthermore, assume that C_f only allows for one computing element at a time to be faulty. Then the network must consist of at least N/2 computing elements.

EXAMPLE I: COMPUTATION OF A MEAN

We want to show the interplay between the definition of a problem and the design of a network of computing elements to solve it. In order to make our point the first example we have chosen is a simplified version of a filtering problem, namely, the computation of the mean of a sequence of numbers.

Problem P:

Scalars numbers $x_1, x_2, x_3, \ldots$ are available at time 1, 2, 3,... etc. and we want to obtain their mean at time 1, 2, 3, ... i.e., we want to have

$$m_i = (\sum_{j=1}^{i} x) / i \qquad (1)$$

for $i = 1, 2, 3, \ldots$

More precisely, we want m_i at time $i + \varepsilon$, with an ε which does not depend on i.

A standard procedure to compute the m_i is the following

$$m_{i+1} = m_i + ((1/(i+1)) (x_{1+1} - m_i) , \qquad (2)$$

with $\qquad m_1 = x_1. \qquad (3)$

We turn now to the hardware implementation of the procedure just defined. We assume that we need the m_i's on line and therefore we must protect ourselves against the occurence of faults. For the case at hand stand-by duplication is not feasible. If a fault occurs in the active computer at time i then all the information about the sequence $\{x_i\}$ up to time i may be lost. We must therefore use either active duplication or allow degradation.

Working Duplication

We use n computing elements $C_1, C_2, \ldots C_n$ in parallel. At time i, the scalar x_i is sent to all n computers and each computing element generates m_i. We use m_i generated by one of the computing elements which is not faulty. The approach is easy to implement but requires a large amount of hardware.

Graceful Degradation

We use a segmented algorithm consisting of four segments AM_1, AM_2, AM_3 and AM_4.

The segment AM_1 is used to compute the mean $\bar{m}_i$ of the x_i with odd i, e.g., $\bar{m}_1 = x_1$, $\bar{m}_3 = (x_1 + x_3)/ 2$, $\bar{m}_5 = (x_1 + x_3 + x_5)/ 3$, etc.

The segment AM_2 is used to compute the mean $\bar{m}_i$ of the x_i with even i, e.g., $\bar{m}_2 = x_2$, $\bar{m}_4 = (x_2 + x_4)/ 2$, $\bar{m}_6 = (x_2 + x_4 + x_6)/ 3$, etc. The determination of the quantities m_i in the segments AM_1 and AM_2 is done by using an iterative procedure essentially identical to the one described in (1).

The segment AM_3 is used to compute the mean m_i for i even, using the formula

$$m_i = (\bar{m}_{i-1} + \bar{m}_i)/ 2 \tag{4}$$

The segment AM_4 is used to compute the mean m_i for i odd, using the formula

$$m_i = \bar{m}_{i+1} + ((i + 1)/(2i))(\bar{m}_i - \bar{m}_{i-1}) \tag{5}$$

The four segments may be implemented on four computing elements C_1, C_2, C_3 and C_4 (see Figure 1).

The advantages of this approach consists in the following:

(i) The rate at which the data is being processed by each computing elements is half the rate at which the x_i's occur.

(ii) If no computing element fails, then the original problem P is solved. If one and only one computing element is faulty, then some approximation of the mean is still available. Let C_f be the class of faults in which at most one computing element is faulty at any one time. Let C_p be the class of problems defined by: compute m_i for all i, or compute m_i for all i odd, or compute m_i for all i even, or compute $\bar{m}_i$ for all i odd or compute $\bar{m}_i$ for all i even. Then, if a fault in C_f occurs, we still solve a problem in C_p.

The use of a suitable segmented algorithm implemented on a network of computing elements allow us to use slower computers and to gain some protection in the occurence of faults.

EXAMPLE II: OPTIMAL CONTROL PROBLEM

The problem we are interested in is a state constrained discrete optimal control problem. A scalar input time invariant system has been chosen, namely

$$z_j = Az_{j-1} + bu_j \quad j = 1,2,\dots,N \tag{6}$$

$$z_0 = \hat{z}_0 \quad \text{given} \tag{7}$$

where for $j = 0, 1, 2,\dots,N$, z_j in E^n is the state of the system at time j, and for $j = 1, 2,\dots,N$ u_j in E is the input at time j. The matrices A and b are real and are of dimension $n\times n$ and $n\times 1$ respectively.

Problem P:

Let T_j, $j = 1, 2,\dots,N$ be convex, closed and bounded subsets of E^n. Find a sequence of inputs $\{\hat{u}_1,\hat{u}_2,\dots,\hat{u}_N\}$, if it exists, such that

$$|\hat{u}_j| \le 1, \quad j = 1,2,\dots,N \tag{8}$$

$$z_j \in T_j \quad j = 1,2,\dots,N \tag{9}$$

$$\sum_{j=1}^{N} (\hat{u}_j)^2 \le \sum_{j=1}^{N} (u_j)^2$$

for all sequences $\{u_1, u_2,\dots,u_N\}$ satisfying (8) and (9).

We now define the class of admissible problems. We suppose that it is admissible to solve problems which differ from P in that 1 of the state constraints is not satisfied i.e., (9) is replaced by

$$z_j \in T_j \text{ for all } j \text{ in } \{1, 2,\dots,N\} \text{ except one.}$$

The admissible class C_p consists therefore of P and P_j where P_j is defined by

Problem P_j:

Identical to Problem P except that (9) is replaced by (10) defined below

$$\begin{aligned} z_j \in T_j \quad &\text{for all } j = 1, 2,\dots,N \\ &\text{except } \quad j = k \end{aligned} \tag{10}$$

$$z_k \in E_n \tag{11}$$

The class C_p consists therefore of P and $P_1, P_2,\dots,P_N$.

We will not enter into the details of the segmented algorithm used for solving problem P. For our purpose, it is sufficient to mention that there exists an algorithm proposed by the author [7] which consists of N+1 segments: one segment is called the MCU and the other N segments are called the SCU's, i.e., SCU-1, SCU-2,...,SCU-N. Each SCU communicates only with the MCU. The computations are organized so that the MCU generates the desired control and each SCU takes care of one state constraint. If one uses N+1 computing elements, the structure of the network will be the one given in Figure 2. If one and only one of the computing element containing an SCU failed, then instead of solving P one will solve P_k, with k corresponding to the SCU-k which is now unavailable. Let C_F be the class of faults which allows at most one computing element to be faulty at any one time. Then if no fault is present P is solved and if a fault in C_f is present, a problem in C_p is solved. An added advantage of using the segmented algorithm is that due to the parallelism of the algorithm, the number of state constraints does not influence the iteration time.

REFERENCES

[1] R.O. Barr and E.G. Gilbert, "Some Efficient Algorithms for a Class of Abstract Optimization Problems Arising in Optimal Control", IEEE Trans. Automatic Control, Vol. AC-14 (1969),pp. 640-652.

[2] J. Cullum, "Finite-Dimensional Approximations of State-Constrained Continuous Optimal Control Problems", SIAM J. Control, Vol. 10 (1972), pp. 649-670.

[3] M. Frank and P. Wolfe, "An Algorithm for Quadratic Programming", U.S. Nav. Res. Logist. Quart., Vol 3 (1956), pp. 95-110.

[4] J.E. Funk and E.G. Gilbert, "Some Sufficient Conditions for Optimality in Control Problems with State Space Constraints", SIAM J. Control, Vol. 8 (1970), pp. 498-504.

[5] D.G. Luenberger, Optimization by Vector Space Methods, Wiley 1969.

[6] G.G.L. Meyer and E. Polak, "A Decomposition Algorithm for Solving a Class of Optimal Control Problems", J. Math. Anal. Appl., Vol. 32 (1970), pp. 118-140.

[7] G.G.L. Meyer, "A Segmented Algorithm for Solving a Class of State Constrained Discrete Optimal Control Problems, IEEE Trans. Automatic Control, Vol. AC-19 (1974), pp.134-136.

[8] E. Polak, Computational Methods in Optimization: A Unified Approach, Academic Press 1971.

[9] A.B. Schwarzkopf, "Optimal Controls for Problems with a Restricted State Space", SIAM J. Control, Vol. 10 (1972), pp. 487-511.

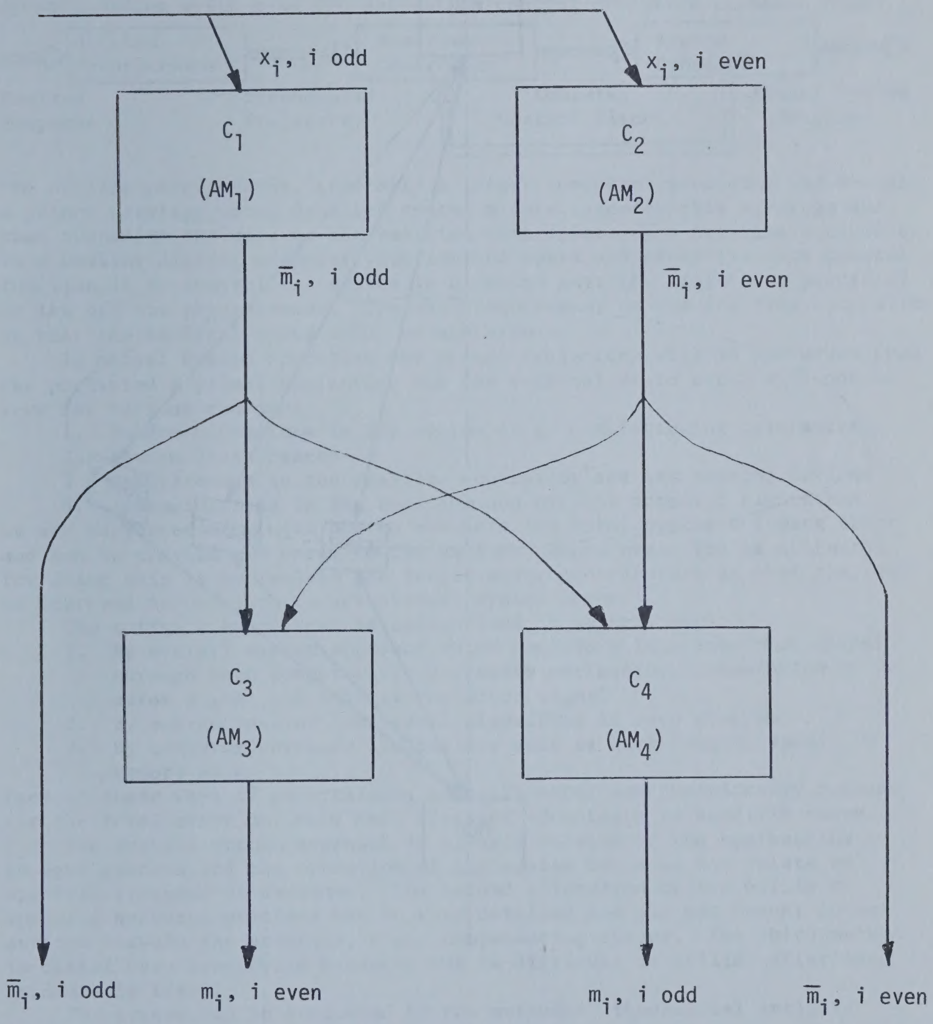

x_i, i odd
x_i, i even
C_1
(AM_1)
C_2
(AM_2)
$\overline{m}_i$, i odd
$\overline{m}_i$, i even
C_3
(AM_3)
C_4
(AM_4)
$\overline{m}_i$, i odd
m_i, i even
m_i, i odd
$\overline{m}_i$, i even

Figure 1

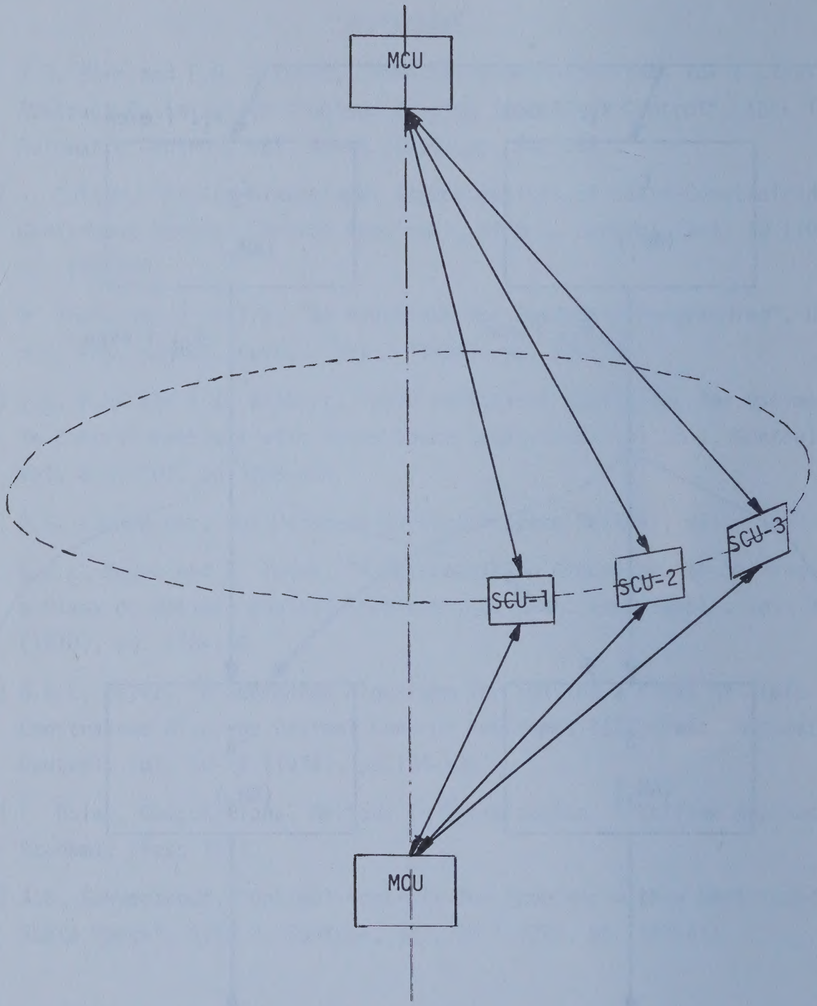

MCU
SCU-1
SCU-2
SCU-3
MCU

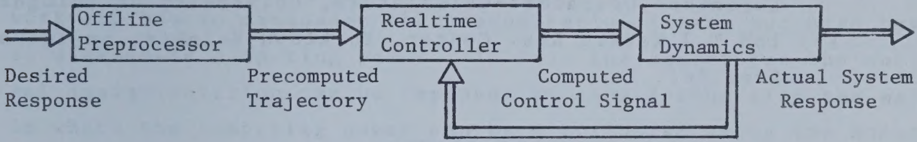
Figure 2

ERROR ANALYSIS IN A MULTIPLE PROCESSOR SOFTWARE SYSTEM

PATRICK D. SHEA and JAMES W. AUSTIN
TRW Systems Group, Redondo Beach, California

SUMMARY

This paper discusses error analysis procedures used in a stochastic dynamic system which uses two sequential control computers as shown below.



The offline preprocessor, generally a larger computer, generates the overall a priori strategy using detailed system models, encodes this strategy and then transmits the data to the realtime controller. The realtime controller is a smaller digital computer with limited speed and capacity. Its general function is to control the system in a region near the trajectory predicted by the offline preprocessor. The main requirement on the realtime controller is that the terminal state error be minimized.

In actual system operation the actual trajectory will be perturbed from the predicted a priori trajectory and the terminal state error will not be zero for various reasons:

1. Random parameters in the system (e.g. manufacturing tolerances)
2. Random disturbances
3. Discreteness in the realtime controller and its sensing devices
4. Approximations in the realtime and offline computer algorithms

We are concerned here with how to evaluate the total system software error and how to ascribe the error to the various components. The major reason for doing this is to isolate the larger error contributors so that they can be improved in order to reduce overall system error.

The software errors can be categorized in several ways:

1. By overall system approach which considers functions that extend through both computers, e.g., state estimation, computation of an error signal, or nulling the error signal
2. By subroutines or individual algorithms in each program
3. By computer hardware limitations such as word length, speed, or memory size

Each of these ways of ascertaining software error can individually account for the total error but each has distinct advantages in specific cases.

The overall system approach is closely related to the engineering thought process and the operation of the system but does not relate to specific programs or routines. The second allocation method points to specific software problems but is very detailed and may not reveal interactions between the programs, e.g., compensating errors. The third method is useful when specifying hardware but is difficult to utilize after the hardware is fixed.

The errors can be evaluated by two methods: theoretical analysis (analytical analysis and computer simulations) and system tests. The theoretical analysis is often more economical and can be started early in the design process before the system is completed but the analysis is not really verified until the system tests have been completed. An additional observation is that it is often difficult to attribute an error to one program or the other because many errors could be reduced in either. Bias errors are often attributed to the off-line program and random dispersion errors to the real-time program.

THE INFLUENCE OF COMPUTER HARDWARE AND COMMUNICATION LINES ON THE COST OF DISTRIBUTED COMPUTER SYSTEMS

L.AMBROSINI GUACCIMANNI(o), G.BUCCI(+)

(o) Istituto di Elettronica and Centro di Studio per l'Interazione Operatore-Calcolatore, University of Bologna

(+) IBM T.J.Watson Res. Center, Yorktown Heights, on leave from (o).

ABSTRACT

A simple model of a Geographically Distributed Computer System is developed, which takes into account both the computer hardware and the communication line cost, in order to investigate under what conditions a distributed system is less expensive than a centralized one.

Some quantitative relations show the weight of the parameters affecting the total system cost.

INTRODUCTION

A Geographically Distributed Computer System (GDCS) is a collection of two or more interconnected computers in different locations which cooperate in order to solve problems. Computer networks lie in this field [R1, K1, K2, I1, S1].

In recent years the growth of these systems has encouraged the development of methods for their design. The proposed methods are of different sorts, but they normally emphasize only one aspect (although very important), that is optimization of the communication network under certain constraints. For instance, given the average traffic emanating from each user location and assuming a high line utilization factor, one chooses the network topology and line capacities which maintain the cost of the system at a minimum [E1, F2, F3, F4].

Chu [C3] and Casey [C1, C2] consider the problem of file copy allocation in an information network, taking into account both the cost of the communication network and that of storing the files at network nodes. Although similar to the warehouse and plant location problems previously approached in [F1, F5], Casey's model is conceptually different since it includes updating traffic, which makes the total traffic in the network dependent on file allocation.

To be more general, the design of a GDCS must involve the invention of some criteria to allocate different kinds of resources [A1]. In this context the term resource allocation includes not only how and where to locate files in a computer network and how to dimension the communication lines, but also how to distribute computing power. In fact the search for the optimal design solution can be improved by considering also the way in which the computing power can be distributed among the nodes of the system when design constraints allow it. This is expected to be useful in dimensioning the system components when the constraints are the geographical locations of the users and the amount of traffic required from each. Obviously, the design must aim at minimizing the overall cost of the system which satisfies the total number of requests (on this subject see, for instance, [D1]).

A terminal located at a great distance from the computer determines a high data communication cost. Therefore it is interesting to implement systems in which, in addition to the central (host) computer, some "satellite" computers are inserted.

Although computer facility decentralization brings about an increase in costs due to the "economies of scale" for computer hardware [K3, S3, O1, L1], satellites can nevertheless satisfy a part of the requests from terminals located close to them; data communication costs will thus be lowered. The hypothesis of obtaining a total system cost reduction through computing power distribution is also justified by data presented in [M1].

Other valid reasons for investigating the dispersion of computing facilities, such as tailoring and reliability, are described in detail in [S4].

In this paper analytical relations are given to evaluate the convenience of a distributed computer system through a rather simplified model in which only computer hardware and data communication costs are considered.

THE MODEL

In this section a very simple distributed computing system model is dealt with.

A user terminal is identified through its position and the number, R, of requests per second it generates. R is assumed constant in time disregarding queueing problems in order to conserve model simplicity. Requests are divided into two categories:

1) Requests which can be satisfied on both a large computer (host) and a small one (satellite).
2) Requests which by their nature can be satisfied only on a large computer.

R_1 and R_2 express the number of requests per unit of time belonging to the first and the second class, respectively. Thus $R = R_1+R_2$.

Following Knight's approach, P denotes the computing power of a system, i.e. the number of standard operations performed per second. It is further assumed that K_1 and K_2 indicate the number of standard operations involved in satisfying type 1 and type 2 requests, respectively. In the following the speed at which the operations in their totality are performed by a whole computing system is considered to be exactly sufficient to satisfy the rate R of requests per second. This is equivalent to assuming that satisfaction of all the request traffic achieves complete saturation of the system.

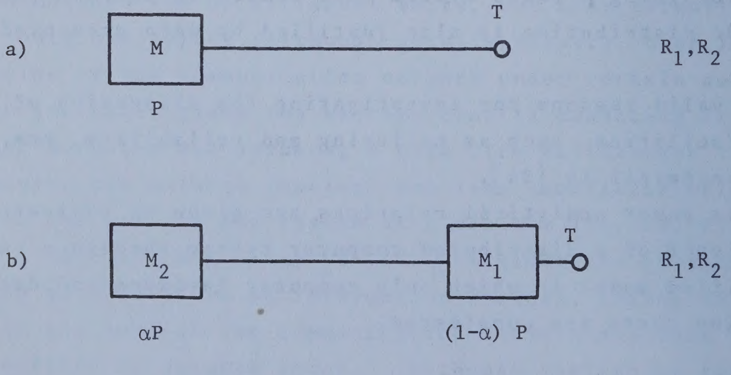


fig. 1

Consider two configurations of a system, as presented in fig.1. We are interested in the problem of finding under what conditions the distributed solution becomes convenient.

On the previous assumption, for the configuration in fig.1 a) it follows that the computer power required to exactly satisfy R requests per second is:

$$R_1K_1 + R_2K_2 = P$$

For the system configuration of fig.1 b), αP and $(1-\alpha)P$ represent the computer power of M_2 (host) and M_1 (satellite), with an arbitrary but fixed value of α. βR_1 gives the portion of the type 1 requests sent to M_2. Thus M_1 has to satisfy only $(1-\beta)R_1$ requests of type 1 per second.

In order to make the behaviour of the two systems the same (i.e., without idle times on any computer) with the same traffic, R, the following relation must hold for the distributed system:

$$\frac{\text{Number of operations to be performed on } M_1}{\text{Number of operations to be performed on } M_2} = \frac{\text{Power of } M_1}{\text{Power of } M_2}$$

which can be put in the form:

$$\frac{(1-\beta)\gamma K_1}{\beta\gamma K_1 + (1-\gamma)K_2} = \frac{1-\alpha}{\alpha} \tag{1}$$

where γ is defined as R_1/R. When $\beta=0$ (minimum communication cost), equation (1) reduces to:

$$\alpha = \frac{1}{1 + \frac{\gamma}{1-\gamma} K}$$

where $K = \frac{K_1}{K_2}$.

By means of (2) it is possible to determine, for a given traffic, what distribution of computing power between the two computers minimizes the communication cost.

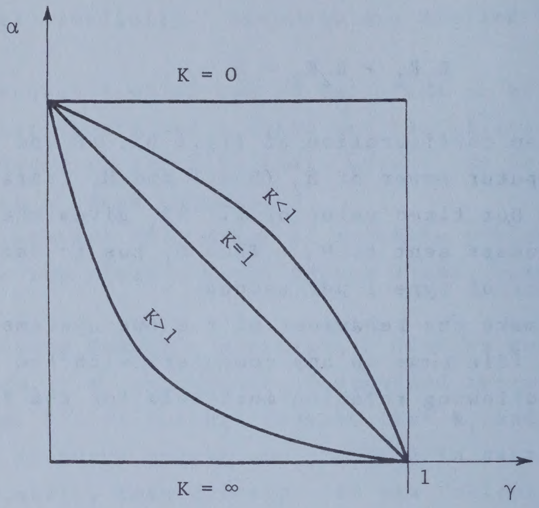


fig. 2

In fig. 2 some curves representing α as a function of γ are drawn for different fixed values of K. With these curves, for given values of R_1, R_2, K_1 and K_2, the value of α that minimizes the communication cost is immediately determined. Obviously, if the hardware cost of computers can be thought of as proportional to their power, this value will also minimize the total cost of the system.

COMPARISON OF COSTS BETWEEN CENTRALIZED AND DECENTRALIZED CONFIGURATION

Many authors (Knight, Solomon, Littrell and others) proved that the rental cost for computer hardware equipment is of the form:

$$C_1 = q_1 P^{\alpha_1}$$

where α_1 is less than 1, showing the economies of scale.

Communication line rental cost can be expressed as a function of bandwidth B in the form:

$$C_2 = q_2 B^{\alpha_2}$$

where α_2 is still less than 1 [G1].

In the configuration of fig.1 a) the cost of the entire system is:

$$\mathrm{Cost}' = q_1 P^{\alpha_1} + q_2 B_o^{\alpha_2} \tag{3}$$

where $B_o = L_1R_1 + L_2R_2$, and L_1 and L_2 are the message lengths of type 1 and type 2 requests, respectively.

In a distributed configuration the total cost can be expressed by:

$$\mathrm{Cost} = q_1((\alpha P)^{\alpha_1} + (1-\alpha P)^{\alpha_1}) + q_2 B^{\alpha_2} \tag{4}$$

where $B = L_2R_2 + \beta L_1R_1$.

For a certain value of total traffic P, when β is ranging between 0 and 1, α shall consequently range between the maximum value $\alpha = 1$ and the minimum value $\alpha = \alpha_{min}$. Since

$$\alpha = \frac{\beta K_1R_1 + K_2R_2}{K_1R_1 + K_2R_2}$$

for $\beta=0$ it follows:

$$\alpha_{min} = \frac{K_2R_2}{P}$$

hence:

$$\beta = \frac{\alpha-\alpha_{min}}{1-\alpha_{min}}$$

then

$$B = L_2R_2 + \frac{\alpha-\alpha_{min}}{1-\alpha_{min}} L_1R_1$$

Now equation (4) can be written in the following form:

$$Cost = q_1 P^{\alpha_1} (\alpha^{\alpha_1} + (1-\alpha)^{\alpha_1} + q \frac{B^{\alpha_2}}{P^{\alpha_1}}$$

where $q = \frac{q_1}{q_2}$.

For fixed values of α_1 and α_2 and of all traffic parameters we may write:

$$Cost = q_1 P^{\alpha_1} (g(\alpha) + H h(\alpha)) \qquad (5)$$

where

$$g(\alpha) = \alpha^{\alpha_1} + (1-\alpha)^{\alpha_1}$$

$$h(\alpha) = (1 + \frac{\alpha-\alpha_{min}}{1-\alpha_{min}} \frac{L_1R_1}{L_2R_2})^{\alpha_2}$$

$$H = \frac{q(L_2R_2)^{\alpha_2}}{P^{\alpha_1}}$$

Finally, it is possible to evaluate the convenience of a decentralized solution compared to the fully centralized one, by considering the ratio:

$$F(\alpha) = \frac{Cost(\alpha)}{Cost(1)} = \frac{g(\alpha) + H h(\alpha)}{g(1) + H h(1)} \qquad (6)$$

For a given traffic and given cost functions, $F(\alpha)$ has the following properties: (a) it is defined in the range $(\alpha_{min},1)$, (b) in this interval it has a downward concavity and (c) $F(1)=1$.

By computing the behaviour of $F(\alpha)$ as a function of α, it is possible to deduce whether and from what starting value of α the decentralized configuration becomes convenient ($F<1$). Such a condition is satisfied if

$$g(\alpha) - g(1) < H(h(1) - h(\alpha))$$

This relation shows how the line cost influence is in opposition to that of the hardware costs (in accordance with intuition), but with weight depending on the H factor which is, by definition, a function of the traffic parameters L_2 and R_2 and of the market situation that determines the values of α_1, α_2, q_1 and q_2. When the values of K_1, K_2, L_1, L_2, q_1, q_2, α_1 and α_2 are given and the needed computing power P has been determined, it is of particular interest to represent α_{min} and $\bar{\alpha}$, characteristic values of α defined as follows, in a graphic form as functions of γ.

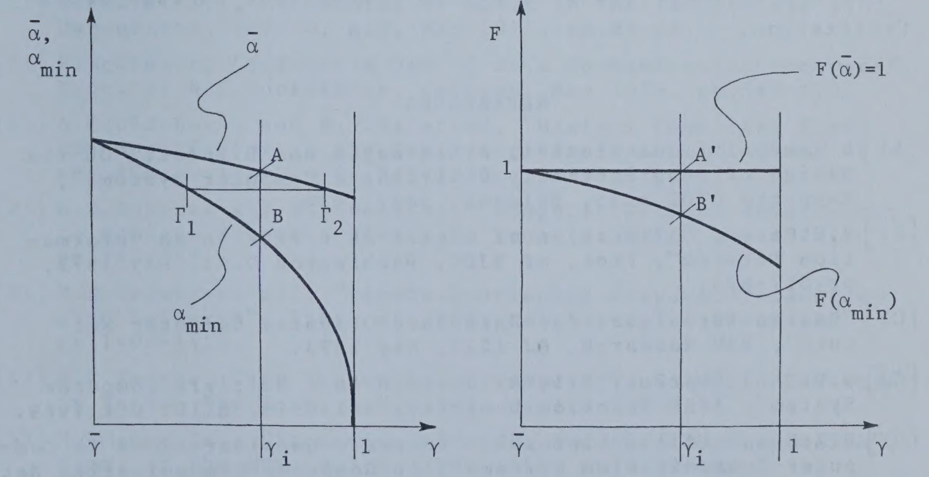
α_{min}: minimum value that can be assigned to α for a given γ.

α_{min} can also be expressed in the form:

$$\alpha_{min} = \frac{1}{1 + \frac{K_1}{K_2}\,\frac{\gamma}{1-\gamma}}$$

$\bar{\alpha}$: value of $\alpha \neq 1$ for which $F(\alpha) = 1$ for the given γ.

The value $\bar{\gamma}$ is defined as the minimum value of γ under which there exists no value of α that verifies the condition $F(\alpha) < 1$. Hence for $\gamma = \bar{\gamma}$ it is $\bar{\alpha} = \alpha_{min}$.



a) b)

fig. 3

In fig. 3 b), $F(\alpha_{min})$ and $F(\bar{\alpha})$ are represented as functions of γ.

The curves of fig. 3 can be used as follows:

a) for each value of γ belonging to $(\bar{\gamma}, 1)$, i.e. for different couples of R_1 and R_2 saturating the power P, the segment AB gives the range of values of α which make the distributed solution more convenient than the centralized one;

b) for a given γ the segment A'B' gives the maximum saving achievable through decentralization;

c) once the value of α, i.e. the power distribution, is chosen, the segment $\Gamma_1\Gamma_2$ gives the excursion allowed for the real traffic to keep the system always in a condition of convenience.

CONCLUSIONS

Economies of scale in computer hardware cost favour system power centralization. To determine whether communication line cost can overturn this situation, some economical relations have been expressed for a simple computer network model.

The study of these relations brought out some conditions to be respected in order to obtain convenience through decentralization.

REFERENCES

[A1] L.Ambrosini Guaccimanni, M.W.Blasgen and G.Bucci, "On the Design of Geographically Distributed Computer Systems", Rapporto CIOC n.25, Bologna, Sept.1974.

[C1] R.G.Casey, "Allocation of Copies of a File in an Information Network", Proc. of SJCC, Washington D.C., May 1972, pp.617-625.

[C2] "Design Techniques for Data Base-Oriented Computer Network", IBM Research, RJ 1222, May 1973.

[C3] W.W.Chu, "Optimal File Allocation in a Multiple Computer System", IEEE Trans.on Computer, vol.C-18, n.10, Oct.1969.

[D1] D.A.Dunn and A.J.Lipinski, "Economic Considerations in Computer-Communication Systems", in Computer-Communication Network, Prentice-Hall, Englewood Cliffs, 1973, pp.371-422.

[E1] L.Esau and K.Williams, "A Method for Approximating the Optimal Network", IBM System Journal, vol.5, n.3, 1966, pp.142-148.

[F1] E.Feldman, F.A.Lehrer and T.L.Ray, "Warehouse Location under Continuous Economies of Scale", Management Science, vol.12, n.9, May 1966, pp.670-684.

[F2] H.Frank and W.Chou, "Topological Optimization of Computer Networks", Proc. of IEEE, vol.60, n.11, Nov.1972, pp.1385-1396.

[F3] H.Frank, R.E.Khan and L.Kleinrock, "Computer Communication Network Design - Experiences with Theory and Practice", Proc. of SJCC, Washington D.C., 1972, pp.255-270.

[F4] H.Frank and I.T.Frisch, "Planning Computer-Communication Network", in Computer Communication Network, 1973, Prentice-Hall, Englewood Cliffs.

[F5] W.D.Frazer, "An Approximate Algorithm for Plant Location under Piecewise Linear Concave Costs", IBM Research Report RC-1875, July 1967.

[G1] R.G.Gould, "Comments on Generalized Cost Expression for Private-Line Communications Channels", IEEE Trans. on Comm. Technology, Com-13, n.3, Sept.1965, pp.374-377.

[I1] IBM System Development Division, "IBM 3650 Retail Store System Introduction", Dept.E01, 1973.

[K1] R.Khan, "Resource-Sharing Computer Communication Networks", Proc. of IEEE, vol.60, n.11, Nov.1972, pp.1397-1407.

[K2] L.Kleinrock, "Analytic and Simulation Methods in Computer Network Design", Proc. of SJCC, Washington D.C., 1970, pp.569-579.

[K3] K.E.Knight, "Changes in Computer Performance", Datamation, Sept.1966, pp.40-54.

[L1] R.F.Littrell, "Economies of Scale in the IBM 360 and 370", Datamation, vol.20, n.3, Mar.1974, pp.83-88.

[M1] P.McGregor, "Effective Use of Data Communication Hardware", Proc. of NCC Conference, Chicago, May 1974, pp.565-575.

[O1] A.E.Oldehoeft and M.H.Halstead, "Maximum Computing Power and Cost Factor in the Centralization Problem", Comm. of the ACM, vol.15, n.2, Feb.1972, pp.94-96.

[R1] L.G.Roberts and B.D.Wessler, "Computer Network Development to Achieve Resource Sharing", Proc. of SJCC, May 1970, pp.543-549.

[S1] M.Schwartz et al., "Terminal-Oriented Computer-Communication Networks", Proc. of IEEE, vol.60, n.11, Nov.1972, pp.1408-1423.

[S2] W.F.Sharpe, "The economics of computer", Columbia University Press, New York, N.Y.,1964, pp.314-322.

[S3] M.Solomon Jr., "Economies of Scale and the IBM System/360", Comm. of the ACM, vol.9, June 1966, pp.435-440.

[S4] D.N.Streeter, "Centralization or Dispersion of Computing Facilities", IBM System Journal, n.3, 1973, pp.283-301.

A MINICOMPUTER FACILITY FOR ADVANCED AUTOMATION RESEARCH

K. S. FU and E. PERSOON
Purdue University, West Lafayette, Indiana 47907

ABSTRACT

At the Laboratory for Pattern Processing and Advanced Automation in Purdue University research in the area of advanced automation is focusing on two topics: the study of manipulators and the study of a system consisting of a mechanical arm and a television camera for visual feedback. The paper describes the equipment and interfaces used for advanced automation research at Purdue University.

1. INTRODUCTION

At the Laboratory for Pattern Processing and Advanced Automation in Purdue University research in the area of advanced automation is focusing on two topics. The first topic deals with the optimal control of manipulators. In order to implement the optimal control strategy, a minicomputer is used in the feedback system. Its task is to obtain data from the manipulator (such as position and velocity of the different joints of the manipulator), to compute in real time the optimal control signals that have to be applied to the motors in the manipulator, and to send those signals to the manipulator. The second research topic studies a system consisting of a mechanical arm and a television camera for visual feedback. The minicomputer is given a task (e.g., pick up a certain object and deposit it somewhere). It uses the camera to locate the object, determines the current position of the arm using the positions of its different joints, and sends control signals to the motors in the arm.

This type of project obviously requires a complex hardware system. The configuration of such a system is constrained by several factors. First, there is a requirement for fast data transfers to and from the peripheral devices. This implies that DMA ports (direct memory access) should be used as much as possible. DMA ports also relieve the CPU of controlling the data transfers and enable the CPU to perform other tasks while a data transfer is in progress. If possible, one should try to perform the data transfer control in the peripheral devices or in a peripheral controller in order to relieve the CPU. Another requirement is to minimize the hardware because it reduces the cost, and more important, because this increases the reliability of the system. The system used at the PPAA laboratory is structured based on those ideas. A block diagram is given in Figure 1.

The minicomputer used is a PDP 11/45 with floating point hardware. Several items are interfaced to the unibus: 32K words core memory, two RK05 moving head discs, a seven-track magnetic tape drive, a 132 character-per-second line printer, a VT05 CRT console, a DR11-B DMA port which is multiplexed into 8 DMA channels, a DR11-C interface which controls the multiplexer, an Optronics film scanner, a 9600-Baud serial interface which connects us with the Purdue University Computing center, a Picture Core Memory used for fast access to picture data, and a cassette tape drive. Four channels of the eight channel multiplexer are used so far to interface the following equipment: a television camera interface, a Ramtek display system, a Perceptronics interface, and a picture display unit. The television camera allows us to digitize images in 127 gray levels. The Ramtek controller can store pictures and generate a TV signal that allows us to display digitized images on a TV monitor. The Perceptronics controller is used to interface the VICARM manipulator to the multiplexer. This controller has A/D converters, D/A converters and digital I/O, and is connected to the multiplexer with 16 data lines going to and 16 data lines coming from the DMA port. One of the channels on the A/D converter is used to provide for speech input. The last item interfaced to the multiplexer is a picture display unit. This unit uses a regular oscilloscope to display digitized images in slow scan. A film camera attached to the oscilloscope records the images.

Many devices in the system have DMA capabilities: magnetic tape drive, RK05 disc drives, drum scanner, and the DR11-B port. The multiplexer eliminates the need for eight DR11-B ports, which are quite expensive. By using a multiplexer, less interfaces to the unibus are needed which improves the performance of the unibus system. The Perceptronics unit can either be controlled by the 11/45 CPU or it can operate independently by executing instructions stored in its processor board. This last capability allows great flexibility in I/O tasks.

In the next sections, we will present the equipment in detail emphasizing the equipment built at Purdue University and emphasizing also the interfacing of the equipment. Section 2 treats the picture processing equipment, and section 3 discusses the Perceptronics equipment and the VICARM manipulator.

2. PICTURE PROCESSING EQUIPMENT

2.1 Optronics Drum Scanner

The Optronics drum scanner is a system P-1000 given in Figure 2. The scanner has a drum which can

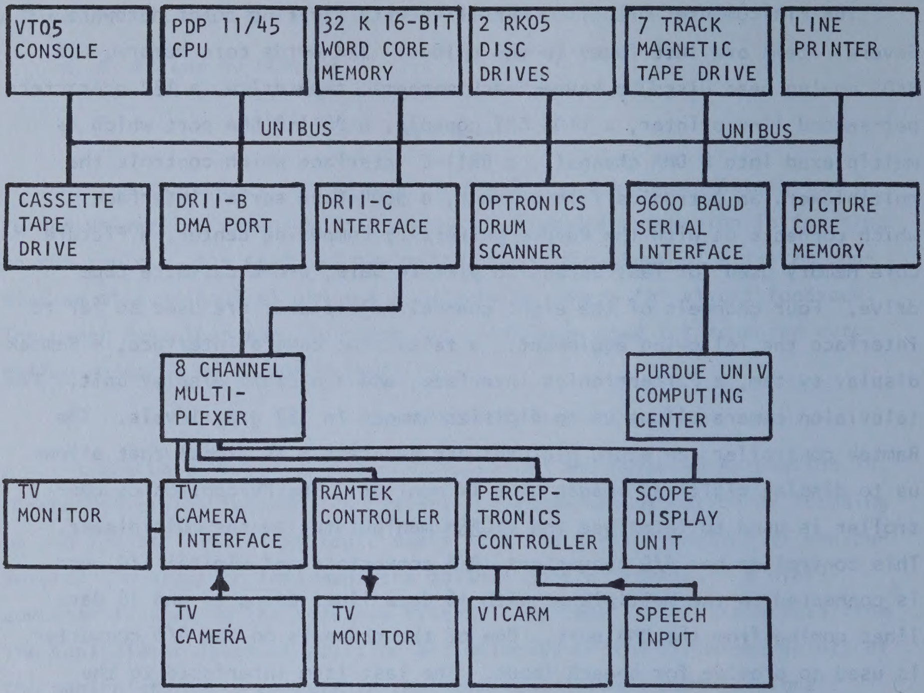

VT05 CONSOLE
PDP 11/45 CPU
32 K 16-BIT WORD CORE MEMORY
2 RK05 DISC DRIVES
7 TRACK MAGNETIC TAPE DRIVE
LINE PRINTER
UNIBUS
UNIBUS
CASSETTE TAPE DRIVE
DR11-B DMA PORT
DR11-C INTERFACE
OPTRONICS DRUM SCANNER
9600 BAUD SERIAL INTERFACE
PICTURE CORE MEMORY
8 CHANNEL MULTI-PLEXER
PURDUE UNIV COMPUTING CENTER
TV MONITOR
TV CAMERA INTERFACE
RAMTEK CONTROLLER
PERCEP-TRONICS CONTROLLER
SCOPE DISPLAY UNIT
TV CAMERA
TV MONITOR
VICARM
SPEECH INPUT

Figure 1. Hardware Configuration at the PPAA Laboratory.

rotate at 2, 4 or 8 revolutions per second. This speed is controlled by the MOTOR selector on the control panel. The drum has a length of 30.75 cm and a circumference of 48 cm. In this drum is an opening of 23 cm by 24 cm in which transparencies can be mounted. From the inside of the drum a spot is focused on the film with selectable sizes of 25, 50, 100 and 200 microns square. Outside the drum on the other side of the film, a photomultiplier detects the light intensity through an aperture with selectable sizes of 25, 50, 100 and 200 microns square. Both the illuminating system and the photomultiplier are mounted on a U-shaped carriage which is controlled by a step motor and steps in increments of 12.5 microns. The output of the photomultiplier is connected to a sample and hold circuit and an A/D converter. This circuitry can sample at different speeds so that a resolution of 25, 50 or 100 microns between sample points can be selected using the RASTER selector. The A/D converter provides 8 bits of resolution (256 levels) in a range of 0-2D or 0-3D optical density. One has the option of feeding the output of the photomultiplier through a logarithmic or a linear amplifier before reaching the sample and hold circuit (see switch on control panel). The U-shaped carriage can be manually positioned using the FORWARD and REVERSE switches. Besides the digitation of film transparencies, the scanner can also scan nontransparent images (printed material) by projecting a light beam using fiber optics on the image from the outside of the drum. Also, color filters are present before the photomultiplier such that color images can be digitized.

2.2 Television Camera Interface

This interface has been built at Purdue University. It accepts regular television signals and allows to digitize those signals into an array of 600 by 500 pixels each with 7 bits of intensity. The interface consists of a sync separator, a video amplifier, sample and hold circuit, an 8 bit A/D converter, a 12 bit counter, an 8 MHZ chock and some control logic. Using this hardware, it samples and digitizes the video signal at one point during one horizontal line of video.

The interface is in a waiting state until the DMA port is programmed. This is accomplished by supplying the following information: the output buffer of the DR11-B is loaded with a 12 bit address, indicating the position on one horizontal line of video where the signal must be digitized, a register is loaded with the number of samples to be taken (between 1 and 250) and another register is loaded with the address in core memory to which the data must be transferred. After the DMA port is programmed, the interface waits for the next vertical sync pulse. During the following horizontal sync pulses the counter is preset with the 12 bit address and starts counting down at a rate of 8 MHZ. When it reaches zero, the video signal is sampled and an A/D conversion is started. At the end of the conversion the digital data are sent to memory. When the interface has transferred the required number of samples, the DMA port becomes inactive and is ready to accept another command. Therefore, one DMA transfer can digitize one vertical line of one video frame. To make this line visible, the video is blanked when the sample is taken and supplied to a video monitor (notice the black line in Figure 4).

The interface can digitize 250 points in 1/60 second (one half frame) or one complete vertical line of 500 points is digitized in 1/30 second (odd and even interlace). Therefore, the time required to digitize 600 by 500 picture points is 20 seconds. The frequency of the 8 MHZ clock in the interface has been adjusted so that no geometrical distortion arises. Due to the DMA transfer from the interface to core memory, there is enough time to format and transfer the data to the disc in real time. Using this basic DMA transfer of one vertical line, a driver has been written that allows to easily digitize images using a television camera. The camera is shown in Figure 3 and is equipped with a small viewfinder. The good quality of this camera allows us to obtain 7 significant bits in the data. Figure 4 shows in the middle the VT05 alphanumeric display which controls the operating system of the 11/45 and on the right the video monitor.

The good qualities of the television camera (sufficient signal to noise ratio, sufficient sensitivity and zoom capability) and the speed of the interface made this interface very attractive as an image input device.

2.3 Ramtek Display System

The Ramtek system has a model GX-100B controller. This controller is connected to a MOS refresh memory from which a video signal is constantly generated. The video signal is then applied to a high-quality video monitor (see Figure 4 on the left). In order to display an image containing 256 by 256 points each with 16 levels of intensity, one needs 256 x 256 x 4 bits of refresh memory. If we define 256 x 256 bits to be one memory plane, the above picture requires 4 memory planes. The system in our laboratory has 12 memory planes which are shared to provide the following display modes: a 256 x 256 display where each point has 256 possible intensity levels (8 memory planes), a 512 x 512 display where each point has 2 intensity levels (4 memory planes) and a 256 x 256 color display consisting of 3 color channels where each channel has 16 intensities (12 memory planes). The first display mode is used for gray level images whereas the second mode is used for graphics applications.

The GX-100B controller is quite complex and resembles a small minicomputer. It has an adequate instruction set (20 instructions) and can work in eight different modes.

Using the complex data mode a driver has been written for displaying gray level images. The driver allows the user to display any part of the picture and also to display pictures at reduced resolution. A Fortran callable subroutine is available that uses the graphic vector mode to draw line sigments. This subroutine is used to display waveforms and functions on the CRT monitor. Another Fortran callable subroutine uses the alphanumeric mode allowing the user to display characters. Figure 4 shows on the monitor on the left a sixteen gray level picture of 256 by 256 points. The picture was digitized using the TV camera interface and represents a portion of the image seen on the monitor on the right.

The Ramtek display system has been a valuable tool for research in picture processing at our laboratory. Figure 5, for example, is a photograph taken from the Ramtek CRT monitor which shows the result of an edge detector, indicated by white lines on the original picture. The efficient and fast operation of the GX-100 B controller provides fast display of any pictorial information.

3. ADVANCED AUTOMATION EQUIPMENT

3.1 Perceptronics Interface Unit

A mechanical manipulator usually has several joints. Each joint requires a motor, potentiometer for positional feedback and a velocity feedback. Therefore, interfacing a manipulator with a minicomputer requires A/D conversion, D/A conversion and digital I/O capabilities. The Perceptronics equipment has those capabilities.

The unit consists of a rack mountable chassis containing a 56 line printed circuit data bus. Several modular cards can be plugged into the data bus. The computer card connects the Perceptronics unit with the the minicomputer through the DR11-B port. It controls the bi-directional transfer of data and interrupts. The clock card contains a clock controlling all data transfers in the Perceptronics unit. Its frequency can be changed under computer control which allows, for example, different sampling rates in the A/D converters. The programmable control card contains 16 words of 16 bit read-write memory. It allows the user to store 16 instructions under computer control. Once this memory is loaded, one can transfer control to this card and the Perceptronics unit will execute the instructions without supervision of the minicomputer. In this mode of operation, data are transferred using DMA.

The three cards mentioned so far form the basic system. Several other cards can be added depending on the particular needs. Outside signals are connected to those cards using edge connectors. The D/A converter card has eight 10 bit D/A converters with a voltage range of ±5 volts. The card has also eight registers to store the digital data to be converted. The A/D converter card consists of a 32 channel analog multiplexer, a sample and hold circuit, and a 10 bit A/D converter plus some control logic. The A/D converter has a range of ±5 volts and has a conversion time of 20 μsec. The digital I/O card provides 16 digital input lines and 16 digital output lines.

The data bus in the Perceptronics contains 16 data lines, 16 address lines, and 5 control lines. The rest of the 56 lines are not important for the next discussion. The 16 address lines are divided in two 8 bit bytes, one byte indicating the card which has to transmit data and the other byte indicating the card that receives data. In each byte, 6 bits contain the address of the card and 2 bits indicate the function the card has to perform. For example, if the A/D card is addressed to transmit data, one might want to perform one of three tasks: decrement the channel address of the multiplexer and start a conversion, increment the channel address and start a conversion, or just start à conversion. In each of those tasks the result of the previous A/D conversion is transmitted on the data lines. The 2 function bits on the address lines allow the user to specify 8 different tasks for each card used in the Perceptronics unit: 4 tasks when the card is addressed to transmit data, and 4 tasks when the card is addressed to receive data. This feature makes operations in the Perceptronics unit very flexible. The five control lines on the data bus are driven by the clock card to control transfers on the bus.

In order to demonstrate the steps required to program the Perceptronics unit, we will take the example of transferring 16 bits of digital data from the minicomputer to the edge connector of the digital I/O card. First, program a DMA transfer of one word. The word contains the 16 bits of digital data. When the DMA transfer is completed, the digital data reside in a data buffer on the computer card. Next, program a DMA transfer of one word containing the address to be loaded on the address lines of the bus. The transmit address refers to the computer card and the receive address to the digital I/O card. When the computer card receives the address, the Perceptronics unit performs the data transfer. The time required by the minicomputer to program the two DMA transfers is approximately 25 microseconds. Therefore, when the minicomputer controls the data transfers in the Perceptronics unit, the maximum data transfer rate is 40.000 16 Bit words per second. When the programmable control card is used to control the data transfers, much higher transfer rates are possible.

3.2 VICARM Manipulator

The manipulator is shown in Figure 6 and its control box is shown in Figure 7. It has been built to simulate the movements of a human arm. The manipulator has six degrees of freedom (joints) where each joint has a motor and a potentiometer for positional feedback. Most joints also have brakes and velocity sensors. The reach of the arm is 50 cm and it has a load capacity of 1.4 kg. Note the two flat cables in the middle of the control box. The top one connects to the Perceptronics interface and the bottom one to the manipulator. LED's on top of the control box signal overtemperature of the motors. A manual control is available (see Figure 7 at the left) which is connected to the control box with the cable shown at the right hand side in Figure 7. This control allows the movement of every joint separately. Individual motors are selected with the switch at the bottom. The shiny dial controls the velocity of the motors. The brakes on each of the motors can be controlled by the switches at the right. An emergency STOP button is also available disconnecting the link between the computer and the manipulator. This button is very useful in the debugging stage of the software that controls the manipulator. The main purpose of the control box is to provide drivers for the motors. It selects either the output of the D/A converters or the signal supplied by the manual control. The selected signal is applied to a current amplifier supplying a current to the motor proportional to the input signal. This arrangement allows an easy control of the torque in the motor and also limits the current in case the motor reaches one of its end points.

4. REMARKS

The hardware configuration used at the PPAA Laboratory has proven to be successful in the areas of pattern processing and advanced automation. We have experienced, however, the limitations of a minicomputer mainly due to the 16 bit word format. Execution is significantly slower than on larger computers. Also, the Fortran compiler provided by DEC produces rather slowly executing code. This forced us to program many critical programs in machine language. Especially in pattern processing, the limitation of addressable memory (28K core) makes fast execution of programs difficult since picture data have to be stored on disc and swapped back and forth during execution.

The slower execution of programs can be avoided by storing the programs at the Purdue University Computing Center and sending the input data and output data over the 9600 Baud line. The storage limitation of 28K will be relieved by the Picture Core Memory now under construction at the laboratory. This memory will provide an access to a maximum of 3.6 billion words with a maximum access time of 5 microseconds. The Picture Core Memory will be implemented with 64K of core memory. The memory is accessed through two 16 bit registers on the unibus providing an effective address of 32 bits. We feel that the complete system, given in Figure 1, will be adequate to perform a large variety of research projects in advanced automation and pattern processing.

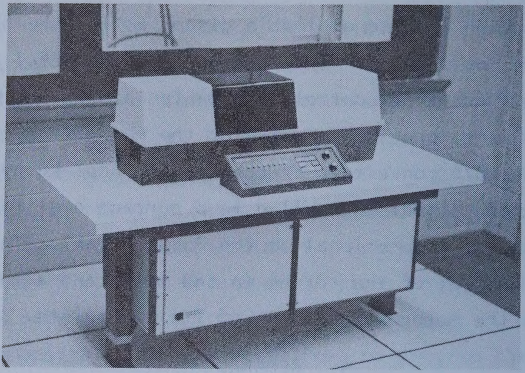


Figure 2

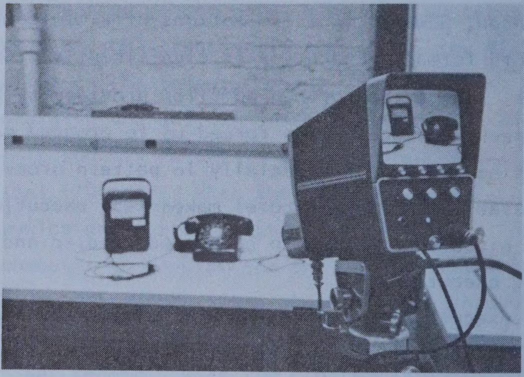


Figure 3

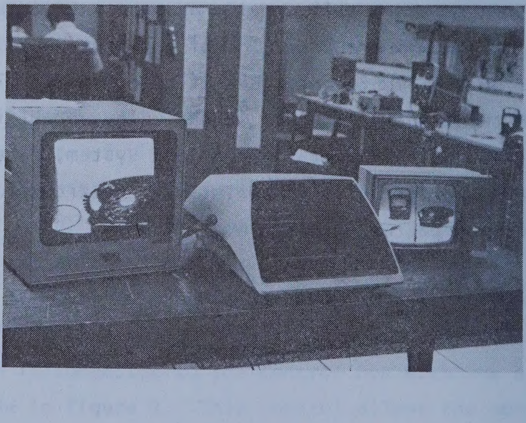


Figure 4

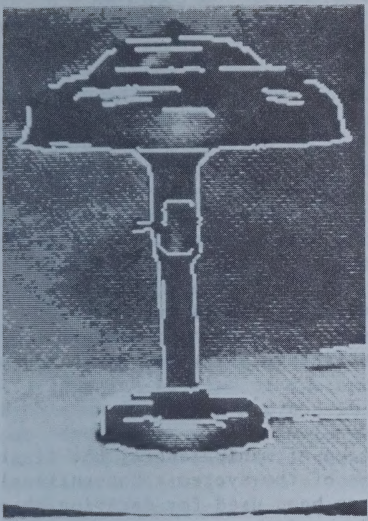


Figure 5

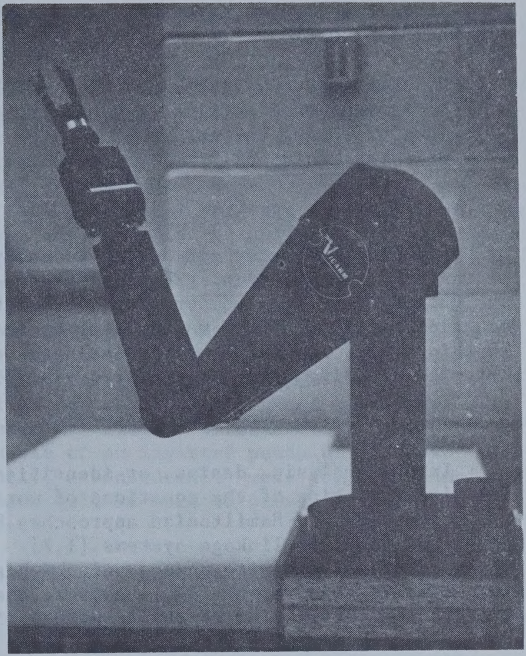


Figure 6

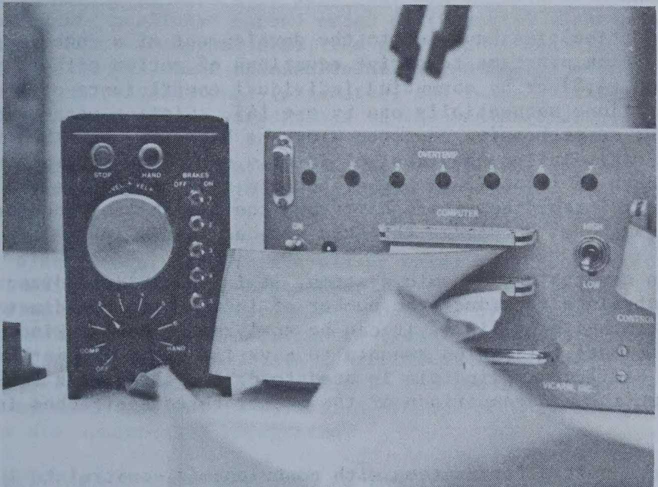

STOP
HAND
BRAKES
OFF ON
OVERTEMP
COMPUTER
HIGH
LOW
CONTROL
VICARM INC.

Figure 7

SOME ALTERNATIVE FORMULATIONS OF MANIPULATOR DYNAMICS FOR COMPUTER SIMULATION STUDIES*

H. HEMAMI, V. C. JASWA, and R. B. MCGHEE
Department of Electrical Engineering
The Ohio State University
Columbus, Ohio 43210

ABSTRACT

A digital computer can assist in analysis, design, or identification of manipulation dynamics. This assistance can be in the form of symbolic derivation, numerical computations, or both, in deriving and solving the equations of motion of a manipulator or in design by trial and error. In this paper, several computer assisted formulations of manipulator dynamics are discussed and compared via illustrative examples.

INTRODUCTION

In the analysis, design, or identification of manipulators, the first step is derivation of the equations of motion of the system. Conventionally, Lagrangian and Hamiltonian approaches have been used for deriving these equations for such linkage systems [1,2]. Both of these approaches permit analysis of systems with open or closed kinematic chains and holonomic or nonholonomic constraints [3]. However, when the number of linkage elements exceeds two, derivation of the equations of motion by hand is time consuming and is prone to human error due to the multiplicity of terms in the equations associated with computation of potential and kinetic energy and to the complexity of transformation to generalized coordinates. This problem becomes even more serious when an attempt is made to obtain the partial derivatives needed in both of these methods and the whole process very soon gets out of hand.

The above difficulties have led to the development of a number of symbolic manipulation programs to derive equations of motion either by the Lagrangian method [4,5] or by computing individual coefficients of the final differential equations sequentially one by one [6]. Similar methods form the basis for the comprehensive computer programs DAMN [7], and IMP [8]. As far as symbolic manipulation programs go, ideally one could program the analytical approach of [9] through use of tensors to characterize the transformation to general coordinates and carry out the Lagrangian procedures, although to the authors' knowledge this has not been done.

For certain constrained dynamic systems, it is possible to directly utilize the constraints to reduce the number of independent coordinates [10]. If a system has many links, it can be analyzed by use of kinematic influence coefficients [11] which amounts to a variation of the methods of [10]. In [12], Hamilton's principle is used to derive difference equations describing the motion. A comparison of the latter three approaches is contained in [6].

A method more suited for systems with nonholonomic constraints is considered in [13], where through direct utilization of the nonholonomic constraints, equations of motion are derived and forces of constraint that do no work are eliminated. This latter approach could perhaps be best explained by a geometrical interpretation of Lagrange's equation; i.e., by projecting

---

*This research was supported by The National Science Foundation under Grant Number ENG74-21664.

Newton's second law onto the moving tangent vector associated with a change in the generalized coordinates [14].

In addition to the above more or less classical approaches to manipulator dynamics, the ready availability of digital computers allows some new approaches to be considered. For example in [15], it is shown that linkage systems can be successfully analyzed by treating each link as a free body connected to other links by elastic joints. As another example, if support reactions are known or measured, then sequential analysis of system dynamics where one link is considered cut at any point in the analysis becomes possible [16]. A comprehensive overview of the status of computer aided linkage system analysis as of 1973 can be found in [17].

The purpose of the present paper is to present what is believed to be a new approach to the analysis of manipulator dynamics based on numerical matrix inversion to obtain constraint forces, and to compare and contrast this approach to some of the earlier computer aided linkage system analysis techniques described above. In order to clarify differences in these various formulations, the simple example of an inverted pendulum as shown in Figure 1 will be used. Subsequently, a considerably more complex example requiring assistance from a computer will be used to provide a more meaningful comparison of these methods.

ALTERNATIVE FORMULATIONS OF MANIPULATOR DYNAMICS

Method I: Free Body Method with Hard Constraints

In this approach, each link of the manipulator is treated as if it were a free body with forces and moments applied at each joint. If the manipulator system is planar, then for each link there are three degrees of freedom and three linear second order equations of motion. If the system is non-planar, then each link possesses six degrees of freedom and is described by six nonlinear second order differential equations (Euler's equations) [2]. In either case, the position of each link is determined by a knowledge of the state of each joint, which may be either rotational (revolute) or translational (prismatic) and these qualities therefore constiture the generalized coordinates of the manipulator system.

Typically, a manipulator joint possesses just one degree of freedom and the torque (revolute joint) or force (prismatic joint) applied along that joint is determined by some type of external control system which may be preprogrammed, manual, or automatic. The other torques and forces (two in the planar case, five in non-planar case) are reaction forces and moments due to motion constraints imposed by the joint structures. In the hard constraint method, these forces are obtained by expressing each link coordinate in terms of joint coordinates (generalized coordinates) and then differentiating twice to obtain a sufficient number of equations to determine all reaction forces and moments. The total set of equations is then solved by numerical matrix inversion. However, only the generalized coordinates are numerically integrated.

Applying the above approach to the system of Figure 1, the three free body equations are

$$m\ddot{x} = F_x \tag{1}$$

$$m\ddot{y} = F_y - mg \tag{2}$$

$$I\ddot{\Theta} = F_x \ell \sin\Theta - F_y \ell \cos\Theta + M \tag{3}$$

The constraint equations for the location of the center of gravity of the link are

$$x = \ell \cos \Theta \tag{4}$$

$$y = \ell \sin \Theta \tag{5}$$

Differentiating twice,

$$\ddot{x} = -\ddot{\Theta}\, \ell \sin \Theta - \dot{\Theta}^2\, \ell \cos \Theta \tag{6}$$

$$\ddot{y} = \ddot{\Theta}\, \ell \cos \Theta - \dot{\Theta}^2\, \ell \sin \Theta \tag{7}$$

Equations (1), (2), (3), (6), and (7) now constitute five equations in the five unknowns $\ddot{x}$, $\ddot{y}$, $\ddot{\Theta}$, F_x, F_y. These equations can be written in the matrix form

$$\begin{bmatrix} m & 0 & 0 & -1 & 0 \\ 0 & m & 0 & 0 & -1 \\ 0 & 0 & I & -\ell \sin\Theta & \ell\cos\Theta \\ 1 & 0 & \ell\sin\Theta & 0 & 0 \\ 0 & 1 & -\ell \cos\Theta & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\Theta} \\ F_x \\ F_y \end{bmatrix} = \begin{bmatrix} 0 \\ -mg \\ M \\ -\dot{\Theta}^2 \ell \cos\Theta \\ -\dot{\Theta}^2 \ell \sin\Theta \end{bmatrix} \tag{8}$$

and then solved by either numerical or symbolic matrix inversion. For the present simple example, symbolic inversion is indicated, but for some more complex examples, numerical inversion has been found to be effective [18]. In either case, some preliminary simplification is usually possible and desirable to reduce the order of the matrix inversion problem. This simplification is generally accomplished by using the second derivatives of the constraint equations to eliminate the superfluous coordinates introduced by the free body assumption. In the present example, this means substituting Eq. (6) and (7) into (1) and (2) to obtain the 3 x 3 problem

$$\begin{bmatrix} I & -\ell \sin\Theta & \ell \cos\Theta \\ m\ell \sin\Theta & 1 & 0 \\ -m\ell \cos\Theta & 0 & 1 \end{bmatrix} \begin{bmatrix} \ddot{\Theta} \\ F_x \\ F_y \end{bmatrix} = \begin{bmatrix} M \\ -m\dot{\Theta}^2 \ell \cos\Theta \\ -m\dot{\Theta}^2 \ell \sin\Theta + mg \end{bmatrix} \tag{9}$$

Method II: Free Body Method with Soft Constraints

The above described Method I involves either numerical or symbolic matrix inversion in order to obtain the second derivatives with respect to time of manipulator joint motions. Khandelwal and Frank [15] have proposed that this inversion can be avoided by a more realistic modelling of the manipulator joints in which it is recognized that the connection between manipulator links is not in fact rigid, but possesses some degree of compliance and damping. Under this assumption, derivation of the differential equations for manipulator motion is greatly simplified, but their order is increased by a factor of three in the planar case and of six in the three-dimensional case since each link is in fact a free body. Despite this complication, it appears that there are cases in which this is a viable alternative approach to computer simulation of linkage system dynamics [15].

With respect to the present illustrative example, referring to Figure 1, in the soft constraint method, the base of the pendulum is set free and given the coordinates x_b, y_b. The "hard" constraint equations (4) and (5) thus become the "soft" constraint equations

$$x = x_b + \ell \cos \Theta \tag{10}$$

$$y = y_b + \ell \sin \Theta \tag{11}$$

with the corresponding "soft" constraint forces

$$F_x = -k_x x_b - k_{\dot{x}} \dot{x}_b \tag{12}$$

$$F_y = -k_y y_b - k_{\dot{y}} \dot{y}_b \tag{13}$$

With these equations, Eq. (1), (2) and (3) become the sixth-order state equations and matrix inversion is avoided at the expense of numerical integration of these equations rather than of the comparatively simple second-order state equation of Method I.

Method III: Generalized Coordinates

If expressions for the total manipulator system kinetic energy and potential energy can be found in terms of independent joint position coordinates (generalized coordinates), then a number of classical approaches including the Lagrangian and Hamiltonian methods can at least in principle be brought to bear on the problem of deriving the system differential equations [2,17]. For example, with respect to the system of Figure 1, the generalized coordinate is Θ and the kinetic energy T is given by

$$T = \frac{1}{2} (m\dot{x}^2 + m\dot{y}^2 + I\dot{\Theta}^2) \tag{14}$$

$$= \frac{1}{2} (m\ell^2\dot{\Theta}^2 \sin^2\Theta + m\ell^2\dot{\Theta}^2 \cos^2\Theta + I\dot{\Theta}^2) \tag{15}$$

$$= \frac{1}{2} (m\ell^2 + I)\dot{\Theta}^2 \tag{16}$$

The potential energy function is

$$V = mg\ell \sin \Theta \tag{17}$$

and the virtual work function is

$$\delta W = Q_\Theta \delta\Theta = M \ \delta\Theta \tag{18}$$

To use the Lagrangian method to obtain the differential equations for any system, it is necessary first to form from the kinetic and potential energy functions the Lagrangian function

$$L = T - V \tag{19}$$

and to then obtain for each generalized coordinate, q_i, the derivative relationship

$$\frac{\partial}{\partial t} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_{q_i} \tag{20}$$

where Q_{q_i} is the coefficient of δq_i in the virtual work function. For the present example, evidently

$$L = \frac{1}{2}\,(m\ell^2 + I)\dot{\Theta}^2 - mg\ell\,\sin\Theta \tag{21}$$

Since Θ is the only generalized coordinate appearing in (21), application of (20) yields the single equation

$$\ddot{\Theta}(m\ell^2 + I) + mg\ell\,\cos\Theta = M \tag{22}$$

or

$$\ddot{\Theta} = \frac{M - mg\ell\,\cos\Theta}{I + m\ell^2} \tag{23}$$

Of course symbolic inversion of (9) yields identical results.

For a system as simple as the above example, the calculations associated with Eq. (18), (19), and (20) are very straightforward and the Lagrangian approach is perhaps the most effective. On the other hand, even small increases in complexity, for example increasing the number of links and joints to three and allowing nonplanar motion [1], complicates these calculations so greatly that the need for computer assistance in both forming the Lagrangian and in carrying out the symbolic differentiation is felt [4,5]. So far as is known to the authors, the differential equations for a six-joint manipulator (the minimal number for arbitrary positioning of a manipulated object) have never been obtained without the aid of a computer in carrying out these two procedures [1,7,8]. It should also be noted that while knowledge of forces of constraint is important to the structural design of manipulators, these forces are suppressed in the classical Lagrangian approach and must be obtained by auxiliary calculations [17]. This is in contrast to either free body method in which the forces of constraint are obtained explicitly during numerical integration of the state equations.

Method IV: Superfluous Coordinates

It is possible to carry out manipulator analysis in a middle ground between Method I and Method III. That is, one can introduce certain superfluous coordinates to simplify writing of the kinetic and potential energy functions without going so far as to set each link completely free. Of course, when such coordinates are introduced, it is also necessary to introduce constraint equations, written in terms of manipulator joint coordinates. Also, it is possible to determine only selected forces and moments of constraint during the calculation in case not all are desired. One way to accomplish such analysis is to make use of Lagrange multipliers to include constraint forces explicitly in either the Lagrangian or Hamiltonian approach. For illustrative purposes, in the following, the Hamiltonian method will be utilized to illustrate the use of Lagrange multipliers with respect to the simple system analyzed above by Methods I, II, and III. A more complex example using the Lagrangian approach with superfluous coordinates and a different method for incorporating constraint forces is presented later in this paper.

To obtain the equations of a manipulator in Hamiltonian form, it is necessary to first compute the generalized momentum vector defined by the column vector [2,8,17]

$$p = \frac{\partial L(x,\dot{x})}{\partial \dot{x}} \tag{24}$$

where x is a vector of coordinates chosen to simplify derivation of the Lagrangian L and which includes as certain of its components the manipulator joint axes. The Hamiltonian is then defined by

$$H(p,x) = p^T\dot{x} - L \tag{25}$$

Since, in general, m superfluous coordinates will be contained in x, it is possible to write an m x 1 vector of constraint equations

$$f(x) = 0 \tag{26}$$

With these definitions, the manipulator equations of motion are given by the system of first order equations

$$\dot{x} = \frac{\partial H}{\partial p} \tag{27}$$

$$\dot{p} = -\frac{\partial H}{\partial x} + [\frac{\partial f}{\partial x}]^T \lambda + \frac{\partial W}{\partial x} \tag{28}$$

where $\partial W/\partial x_i$ is the coefficient of δx_i in the virtual work function and λ is the m x 1 vector of unknown Lagrange multipliers. Evidently, some auxiliary equations are needed to evaluate λ. These can be obtained by noting from Eq. (26) that since $f(x) = 0$, it follows that

$$\dot{f} = (\partial f/\partial x)\dot{x} = 0 \tag{29}$$

Consequently, it also follows that if

$$z = (x,p)^T \tag{30}$$

then

$$\dot{f} = (\partial f/\partial z)\dot{z} = 0 \tag{31}$$

and

$$\ddot{f} = \frac{d}{dt}[(\partial f/\partial z)\dot{z}] = 0 \tag{32}$$

It is this last equation which generally yields the desired forces of constraint. This will be illustrated with the previously employed inverted pendulum system.

For the example of Figure 1, it is appropriate to choose

$$x = (x_1,x_2,x_3)^T = (x,y,\Theta)^T \tag{33}$$

Then, from Eq. (14), (17), and (19),

$$p = (m\dot{x}, m\dot{y}, I\dot{\Theta})^T \tag{34}$$

and

$$H = \frac{1}{2m}p_1^2 + \frac{1}{2m}p_2^2 + \frac{1}{2I}p_3^2 + mg\,x_2 \tag{35}$$

Thus

$$\dot{x} = \frac{\partial H}{\partial p} = (\frac{p_1}{m}, \frac{p_2}{m}, \frac{p_3}{I})^T \tag{36}$$

From Eq. (4) and (5), the vector constraint equation is

$$f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} = \begin{pmatrix} x - \ell\cos\Theta \\ y - \ell\sin\Theta \end{pmatrix} = 0 \tag{37}$$

so

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \partial f_1/\partial x \\ - - - - \\ \partial f_2/\partial x \end{bmatrix} = \begin{bmatrix} 1 & 0 & \ell\sin\Theta \\ 0 & 1 & -\ell\cos\Theta \end{bmatrix} \tag{38}$$

and from Eq. (18) and (28)

$$\dot{p} = \begin{pmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dot{p}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -mg \\ 0 \end{pmatrix} + [\frac{\partial f}{\partial x}]^T \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ M \end{pmatrix} \tag{39}$$

or

$$\dot{p} = \begin{pmatrix} m\ddot{x} \\ m\ddot{y} \\ I\ddot{\Theta} \end{pmatrix} = \begin{pmatrix} \lambda_1 \\ -mg + \lambda_2 \\ \lambda_1\,\ell\sin\Theta - \lambda_2\,\ell\cos\Theta \end{pmatrix} \tag{40}$$

At this point it can be noted that if $\lambda_1 = F_x$ and $\lambda_2 = F_y$ then Eq. (40) reduces to Eq. (1), (2), and (3). This verifies that the analysis up to this point is correct, but does not yield values for these two unknowns. Instead, these quantities must be obtained from Eq. (32). To obtain this equation, it is necessary to first compute

$$\dot{f} = [\frac{\partial f}{\partial x}]\dot{x} = \begin{pmatrix} \dot{x} + \dot{\Theta}\ell\sin\Theta \\ \dot{y} - \dot{\Theta}\ell\cos\Theta \end{pmatrix} = 0 \tag{41}$$

and to then differentiate again to obtain

$$\ddot{f} = \begin{pmatrix} \ddot{x} + \ddot{\Theta}\ell\sin\Theta + \dot{\Theta}^2\ell\cos\Theta \\ \ddot{y} - \ddot{\Theta}\ell\cos\Theta + \dot{\Theta}^2\ell\sin\Theta \end{pmatrix} = 0 \tag{42}$$

Substituting the relationships of Eq. (40) for the second derivatives appearing in Eq. (42) yields the desired two simultaneous equations in the components of λ. For more complex problems it would be necessary to invert these equations numerically. However, for the present simple example, analytic inversion is possible and yields the result

$$\lambda = \begin{pmatrix} \lambda_1 \\ \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} -m\ddot{\Theta}\ell \sin\Theta - m\dot{\Theta}^2 \ell\cos\Theta \\ \\ mg + m\ddot{\Theta}\ell \cos\Theta - m\dot{\Theta}^2 \ell \sin\Theta \end{pmatrix} \tag{43}$$

Finally, substitution of this equation into Eq. (40) produces Eq. (23), thereby validating all of the above calculations.

COMPARISON OF METHODS

Each of the above approaches has certain advantages and disadvantages for computer implementation. Methods I and II are the easiest to formulate, but numerical integration of the resulting equations is comparatively difficult since it is necessary to deal with either a large numerical matrix inversion problem (Method I) or else with a considerably increased system order, perhaps with widely separated eigenvalues (Method II). Method III is the one most often used for comparatively simple problems, but the analytic manipulations required become effectively impossible to carry out without the aid of a computer for systems as complex as the typical manipulator. Method IV organizes the numerical evaluation of the constraint forces in a way which reduces the size of the matrix inversion problem associated with Method I. The complexity of the manipulations required to attain this reduction depends in general on the number of superfluous coordinates used and may or may not be justified in specific instances. In the following example, yet another variant of Method IV is presented and contrasted to application of Method III with respect to a three joint manipulator.

ANALYSIS OF A THREE JOINT MANIPULATOR

Method III: Computer Assisted Classical Lagrangian Approach

The system shown in Figure 2 is identical to the one presented in [1] except for a minor change in coordinates. This system was analyzed using Dillon's computer based system [4,5] both to obtain the Lagrangian and to carry out the necessary symbolic differentiations. The results of this calculation are listed in Appendix A in which the Lagrangian appears as Eq. (A-1) and the three derivative relationships implied by Eq. (20) are presented as Eq. (A-2) through (A-4).

Examination of the equations of Appendix A leads the authors to believe that while such calculations could certainly be carried out without the assistance of a computer, the probability of error would be quite high. Moreover, it should be noted that even these equations cannot in fact be integrated without solving (A-2) through (A-4) by symbolic or numerical matrix inversion to obtain the three joint angle accelerations explicitly. Symbolic inversion produces expressions which are considerably more complicated [1] with an attendant greater opportunity for error.

Method IV: Superfluous Coordinates with Numerical Matrix Inversion

The essence of this method is to choose coordinates which make it easy to write the kinetic and potential energy functions for the manipulator system and to then constrain these coordinates by calculation of constraint forces by numerical matrix inversion. Thus it is not necessary to set each link entirely free as in the simple inverted pendulum example, but merely to free it to the extent needed to allow easy derivation of the Lagrangian. Specifically, referring to Figure 2, the two inertial angular velocities that the third link possesses, ϕ_3 due to its own motion and ϕ_1 due to the motion of link 1, can be decomposed into three orthogonal components along a coordinate system fixed to link 3, and given by $\dot\phi_3$, $\dot\phi_1 \cos \phi_3$, and $\dot\phi_1 \sin \phi_3$. Since the center of gravity of link 3 moves in 3 dimensions, it has 3 linear velocities $\dot x_3$, $\dot y_3$, and $\dot z_3$. As a result, since all of the above velocities are with respect to the inertial reference frame (x,y,z), the Lagrangian for this link may be directly written as

$$L_3 = 1/2\ m_3\dot x_3^2 + 1/2\ m_3\dot y_3^2 + 1/2\ m_3\dot z_3^2 + 1/2\ J_{3yy}\dot\phi_3^2 + 1/2\ J_{3xx}(\dot\phi_1 \sin\phi_3)^2$$

$$+ 1/2\ J_{3zz}(\dot\phi_1 \cos\phi_3)^2 - m_3 g z_3 \tag{44}$$

where the three rotational inertias are about body-fixed axes and are hence time-invariant. If one were to imagine link 3 as having all of the degrees of freedom ascribed to it in the Lagrangian, then as shown in Figure 3, three forces, F_{x3}, F_{y3}, and F_{z3}, and a moment of constraint, M_3, must be regarded as acting upon the link.

To obtain the equations of motion for link 3, one uses the Lagrange

$$\frac{d}{dt}\left(\frac{\partial L_3}{\partial \dot q_3}\right) - \frac{\partial L_3}{\partial q_3} = Q_{q3} \tag{45}$$

where q_3 is each of the coordinates used to express the Lagrangian and Q_{q3} is the generalized force acting upon the link and may stand for moments as well as forces. It is found as follows and herein lies part of the simplification afforded by the addition of superfluous coordinates.

$$Q_{q3} = \Sigma\ F_j \frac{\partial q_j}{\partial q_3} \tag{46}$$

where q_j is the value of coordinate along which each force F_j acts measured at the point of application. For example, the force F_{x3} acts along the x_3 coordinate at $q_{x3} = x_3 - k_3\cos\phi_3\cos\phi_1$; the force F_{y3} acts along y_3 at $q_{y3} = y_3 - k_3\cos\phi_3\sin\phi_1$; the torque M_3 acts about ϕ_1 and so on. Now, employing Eq. (45) with $q_3 = x_3$ produces the result

$$m\ddot x_3 = Q_{x3} = \Sigma\ F_j \frac{\partial q_j}{\partial x_3}$$

$$= F_{x3}\frac{\partial}{\partial x_3}(x_3 - k_3\cos\phi_3\cos\phi_1) + F_{y3}\frac{\partial}{\partial x_3}(y_3 - k_3\cos\phi_3\sin\phi_1)$$

$$+ F_{z3}\frac{\partial}{\partial x_3}(z_3 - k_3\sin\phi_3) + M_3\frac{\partial \phi_1}{\partial x_3} + T_3\frac{\partial \phi_3}{\partial x_3}\ . \tag{47}$$

Similar relations can be written for y_3 and z_3 resulting finally in the familiar Newton's laws of motion:

$$m_3\ddot x_3 = Fx_3$$

$$m_3\ddot{y}_3 = Fy_3 \tag{48}$$

$$m_3\ddot{z}_3 = m_3 g = Fz_3$$

Further application of Eq. (45) to this link for ϕ_3 and ϕ_1 respectively produces

$$\ddot{\phi}_3 J_{3yy} + \dot{\phi}_1^2 \sin\phi_3 \cos\phi_3 (J_{3zz} - J_{3xx})$$

$$= -F_{z3}k_3\cos\phi_3 + F_{x3}k_3\sin\phi_3\cos\phi_1 + F_{y3}k_3\sin\phi_3\sin\phi_1 + T_3 \tag{49}$$

$$\ddot{\phi}_1\{J_{3zz}\cos^2\phi_3 + J_{3xx}\sin^2\phi_3\} + 2\dot{\phi}_1\dot{\phi}_3\sin\phi_3\cos\phi_3(J_{3xx} - J_{3zz})$$

$$= F_{x3}k_3\cos\phi_3\sin\phi_1 - F_{y3}k_3\cos\phi_3\cos\phi_1 + M_3 \tag{50}$$

Similarly, for the middle and first links one obtains the equations listed in Appendix B. The ease with which they are derived must be contrasted with the fact that the equations listed in Appendix A had to be generated on a computer.

At this point in the analysis, 13 equations in 21 unknowns comprised of the second derivatives of the coordinates used and the forces and moments of constraints have been obtained. These appear as Eqs. (B-4) through (B-16) in Appendix B. In addition to these, eight equations of constraint are found by tying the links together. They appear after double differentiation as Eqs. (B-17) through (B-24). Thus, taken all together, Eq. (B-4) through (B-24) constitute the desired set of 21 linear simultaneous equations in the unknown link coordinate acceleration and forces of constraint arising from the influence of gravity forces, acceleration forces, and applied torques acting upon the manipulator links. These equations are to be inverted numerically during the course of computer integration of the joint acceleration and velocities. Since the matrix to be inverted is quite sparse, if desired, its order can be reduced somewhat from the 21 x 21 size implied by Appendix B by preliminary analytic manipulations similar to those used in obtaining Eq. (9) from Eq. (8).

Comparison of Solution Times

Eqs. (B-4) through (B-24) and Eqs. (A-2) through (A-4) were simulated on an IBM 370/165 computer. The manipulator joint trajectories were identical when the same applied torque functions were utilized, thereby tending to confirm the validity of both simulations. The results of the simulation also showed the price that must be paid when using the superfluous coordinate method in compensation for the ease of derivation it affords. In simulating an actual period of manipulator motion of two seconds, the execution time using the direct Lagrangian equations was only 0.54 seconds as compared to 7.18 seconds required by the other method. Of course, the computing time for Method IV could be reduced somewhat if the order of the matrix inversion problem were to be first reduced by analytic manipulations as described above, or if special inversion routines were to be used to take advantage of the sparseness of this matrix. Likewise, the computing time of this method would also change if either more or less superfluous coordinates were to be used. There is thus a considerable degree of arbitrariness in this comparison of execution times. The authors feel that more comparisons of this sort are needed in order to obtain a better assessment of the relative advantages and disadvantages of the methods proposed herein and would welcome similar results from others interested in computer simulation of manipulator dynamics.

CONCLUSIONS

In this paper some alternative formulations of manipulator dynamics that are of potential use in computer-aided system analysis have been discussed. It was found that the use of numerical matrix inversion to solve for forces and moments of constraint can simplify derivation of the equations of motion for such systems at the expense of increasing computer solution times. Additional interesting problems are those of identification of system parameters from input-output observations and design of stable trajectories for manipulation and linkage systems. Unfortunately, at present it appears that most of the conventional feedback design techniques can only be applied to systems that are in Lagrangian form involving only independent state variables. More work is necessary to establish design techniques that could directly be applied to any of the four methods discussed above or combinations thereof. In any event, however, in the computer simulation of manipulator dynamics, it is valuable to have more than one method of obtaining the system dynamic equations since if solution of the various sets of equations obtained by different methods yields identical results, much greater confidence can be placed in the correctness of the simulation programs.

REFERENCES

1. Kahn, M. E., and Roth, B., "The Near Minimum-Time Control of Open-Loop Articulated Kinematic Chains," Trans. ASME, Vol. 93, Series G, No. 3, J. of Dynamic Systems, Measurement, and Control, Sept. 1971, pp. 164-172.

2. Whittaker, E. T., A Treatise on the Analytical Dynamics of Particles and Rigid Bodies, Dover Publications, New York, 1937.

3. Chace, M. A., "Analysis of the Time Dependence of Multi-Freedom Systems in Relative Coordinates," J. of Engr. for Industry, Trans. ASME, Series B, Vol. 19, No. 1, February 1967, p. 111.

4. Dillon, S. R., Computer Assisted Equation Generation in Linkage Dynamics, Ph.D. dissertation, The Ohio State University, Columbus, Ohio, August 1973.

5. Dillon, S. R., and Hemami, H., "Automated Equation Generation and Its Application to the Problems in Control," Proc. of 15th JACC, Austin, Texas, pp. 575-580, 1974.

6. Chace, M. A., and Bayazitoglu, Y. O., "Development and Application of a Generalized d'Alemebert Force for Multi Freedom Mechanical Systems," J. of Engr. for Industry, Trans. ASME, Feb. 1971, pp. 317-327.

7. Smith, D. A., et al., "The Automatic Generation of a Mathematical Model for Machinery Systems," Trans. ASME, J. of Engr. for Industry, May 1973, pp. 629-635.

8. Sheth, P. N., and Uicker, Jr., J. J., "IMP (Integrated Mechanisms Program), A Computer-Aided Design Analysis System for Mechanisms and Linkage," Trans. ASME, J. of Engr. for Industry, May 1972, pp. 454-464.

9. Maros, D., and Orlandea, N., "Contribution to the Determination of the Equations of Motion for Multidegree of Freedom Systems," *J. of Engr. for Industry, Trans. of ASME*, Feb. 1971, pp. 191-195.

10. Bonham, W. E., "Calculating the Response of a Four-Bar Linkage System," ASME paper No. 70-Mech-60, 1970.

11. Benedict, C. E., and Tesar, D., "Optimal Torque Balance for a Complex Stamping and Indexing Machine," ASME paper No. 70-Mech-82, 1970.

12. Vance, J. M., and Sitchin, A., "Derivative of First-Order Difference Equations for Dynamical Systems by Direct Application of Hamilton's Principle," *J. of Applied Mechanics*, Vol. 37, *Trans. ASME*, Vol. 92, No. 2, pp. 276-278, June 1970.

13. Kane, T. R., "Dynamics of Nonholonomic Systems," *J. of Appl. Mech., Trans. ASME*, Dec. 1961, pp. 574-578.

14. Beers, B. L., "Geometric Nature of Lagrange's Equations," *Am. J. of Physics*, Nov. 1972, pp. 1636-1640.

15. Khandelwal, B. M., and Frank, A. A., "On the Dynamics of an Elastically Coupled Multi-Body Biped Locomotion Model," *Proc. of 1974 JACC*, Asutin, Texas, June 1974.

16. Vukobratovic, M., and Stepanenko, J., "Mathematical Models of General Anthropomorphic Systems," *Math. Biosciences*, Vol. 17, pp. 191-242, 1973.

17. Paul, B.,"Analytical Dynamics of Mechanisms--A Computer Oriented Overview," to be published in *Mechanisms and Machine Theory, The J. of IFTOMM*, The International Federation for the Theory of Machines and Mechanisms. Also available as Univ. of Pennsylvania, Dept. of Mechanical Engr. and Applied Mechanics, MEAM Report 74-1, January 1974.

18. Gupta, S., *Estimation of Lower Limb Joint Forces and Moments Using On-Line Measurements and Computations*, M.S. thesis, The Ohio State University, Columbus, Ohio, March 1975.

APPENDIX A: CLASSICAL LAGRANGIAN ANALYSIS OF THREE JOINT MANIPULATOR

$$
\begin{aligned}
L = {} & 1/2\, \dot{\phi}_1^2 \{J_{1zz} + J_{2xx}\sin^2\phi_2 + J_{2zz}\cos^2\phi_2 + J_{3zz}\cos^2\phi_3 + J_{3xx}\sin^2\phi_3\} \\
& + \dot{\phi}_1^2 \{m_3 a_1 a_2 \cos\phi_2 + m_2 a_1 k_2 \cos\phi_2 + m_3 a_1 k_3 \cos\phi_3 + m_3 a_2 k_3 \cos\phi_2 \cos\phi_3\} \\
& + 1/2\, \dot{\phi}_1^2 \{m_1 k_1^2 + m_2 k_2^2 \cos^2\phi_2 + m_3 k_3^2 \cos^2\phi_3 + m_2 a_1^2 + m_3 a_1^2 \\
& + m_3 a_2^2 \cos^2\phi_2\} + 1/2\, \dot{\phi}_2^2 \{J_{2yy} + m_2 k_2^2 + m_3 a_2^2\} + \dot{\phi}_2 \dot{\phi}_3 m_3 a_2 k_3 \cos(\phi_3 - \phi_2) \\
& + 1/2\, \dot{\phi}_3^2 \{J_{3yy} + m_3 k_3^2\} - m_2 g k_2 \cos\phi_2 - m_3 g(a_2 \cos\phi_2 + k_3 \cos\phi_3) \qquad \text{(A-1)}
\end{aligned}
$$

$$\begin{aligned}
0 = T_1 &+ \ddot{\phi}_1 \{- J_{1zz} - J_{2xx}\sin^2\phi_2 - J_{2zz}\cos^2\phi_2 - J_{3zz}\cos^2\phi_2 - J_{3xx}\sin^2\phi_2 \\
&- 2m_3a_1a_2\cos\phi_2 - 2m_2a_1k_2\cos\phi_2 - 2m_3a_1k_3\cos\phi_3 - 2m_3a_2k_3\cos\phi_2\cos\phi_3 \\
&- m_1k_1^2 - m_2k_2^2\cos^2\phi_2 - m_3k_3^2\cos^2\phi_3 - m_2a_1^2 - m_3a_1^2 - m_3a_2^2\cos^2\phi_2\} \\
&+ \dot{\phi}_1\dot{\phi}_2 \{2J_{2zz}\cos\phi_2\sin\phi_2 - 2J_{2xx}\sin\phi_2\cos\phi_2 + 2m_2a_1k_2\sin\phi_2 + 2m_3a_1a_2\sin\phi_2 \\
&+ 2m_2k_2^2\cos\phi_2\sin\phi_2 + 2m_3a_2^2\cos\phi_2\sin\phi_2 + 2m_3a_2k_3\sin\phi_2\cos\phi_3\} \\
&+ \dot{\phi}_1\dot{\phi}_3 \{2J_{3zz}\cos\phi_2\sin\phi_3 - 2J_{3xx}\sin\phi_3\cos\phi_3 + 2m_3a_1k_3\sin\phi_3 \\
&+ 2m_3a_2k_3\cos\phi_2\sin\phi_3 + 2m_3k_3^2\sin\phi_3\cos\phi_3\} \qquad \text{(A-2)}
\end{aligned}$$

$$\begin{aligned}
0 = T_2 &- m_2gk_2\cos\phi_2 - m_3ga_2\cos\phi_2 - m_3gk_3\cos\phi_3 \\
&+ \ddot{\phi}_2 \{- J_{2yy} - m_2k_2^2 - m_3a_2^2 - m_3a_2k_3\cos(\phi_3-\phi_2)\} \\
&+ \ddot{\phi}_3 \{- J_{3yy} - m_3k_3^2 - m_3a_2k_3\cos(\phi_3-\phi_2)\} \\
&+ \dot{\phi}_1^2 \{J_{2xx}\cos\phi_2\sin\phi_2 - J_{2zz}\cos\phi_2\sin\phi_2 - J_{3zz}\cos\phi_3\sin\phi_3 + J_{3xx}\cos\phi_3\sin\phi_3 \\
&\quad - m_3a_1a_2\sin\phi_2 - m_2a_1k_2\sin\phi_2 - m_3a_1k_3\sin\phi_3 \\
&\quad - m_2k_2^2\cos\phi_2\sin\phi_2 - m_3k_3^2\cos\phi_3\sin\phi_3 - m_3a_2^2\cos\phi_2\sin\phi_2 \\
&\quad - m_3a_2k_3\sin(\phi_2+\phi_3)\} - \dot{\phi}_2^2m_3a_2k_3\sin(\phi_3-\phi_2) + \dot{\phi}_3^2m_3a_2k_3\sin(\phi_3-\phi_2) \qquad \text{(A-3)}
\end{aligned}$$

$$\begin{aligned}
0 = T_3 &- m_3gk_3\cos\phi_3 - \ddot{\phi}_2m_3a_2k_3\sin\phi_3 + \ddot{\phi}_3 \{-J_{3yy} - m_3k_3^2\} \\
&+ \dot{\phi}_1^2 \{- J_{3zz}\sin\phi_3\cos\phi_3 + J_{3xx}\sin\phi_3\cos\phi_3 - m_3a_1k_3\sin\phi_3 - m_3a_2k_3\cos\phi_2\sin\phi_3 \\
&- m_3k_3^2\sin\phi_3\cos\phi_3 - \dot{\phi}_2^2\, m_3a_2k_3\sin(\phi_3-\phi_2)\} \qquad \text{(A-4)}
\end{aligned}$$

APPENDIX B: LAGRANGIAN ANALYSIS OF A THREE JOINT MANIPULATOR USING SUPERFLUOUS COORDINATES

$$L_1 = 1/2\, m_1\dot{x}_1^2 + 1/2\, m_1\dot{y}_1^2 + 1/2\, J_{1zz}\dot{\phi}_1^2 - m_1 g z_1 \tag{B-1}$$

$$\begin{aligned} L_2 = {} & 1/2\, m_2\dot{x}_2^2 + 1/2\, m_2\dot{y}_2^2 + 1/2\, m_2\dot{z}_2^2 + 1/2\, J_{2yy}\dot{\phi}_2^2 + 1/2\, J_{2zz}\{\dot{\phi}_1\cos\phi_2\}^2 \\ & + 1/2\, J_{2xx}\{\dot{\phi}_1\sin\phi_2\}^2 - m_2 g z_2 \end{aligned} \tag{B-2}$$

$$\begin{aligned} L_3 = {} & 1/2\, m_3\dot{x}_3^2 + 1/2\, m_3\dot{y}_3^2 + 1/2\, m_3\dot{z}_3^2 + 1/2\, J_{3yy}\dot{\phi}_3^2 \\ & + 1/2\, J_{3zz}\{\dot{\phi}_1\cos\phi_3\}^2 + 1/2\, J_{3xx}\{\dot{\phi}_1\sin\phi_3\}^2 - m_3 g z_3 \end{aligned} \tag{B-3}$$

$$m_1\ddot{x}_1 = F_{x1} - F_{x2} \tag{B-4}$$

$$m_1\ddot{y}_1 = F_{y1} - F_{y2} \tag{B-5}$$

$$\begin{aligned} J_{1zz}\ddot{\phi}_1 = {} & T_1 - M_2 + F_{x1}k_1\sin\phi_1 - F_{y1}k_1\cos\phi_1 + F_{x2}(a_1-k_2)\cos\phi_1 \\ & - F_{y2}(a_1-k_1)\sin\phi_1 \end{aligned} \tag{B-6}$$

$$m_2\ddot{x}_2 = F_{x2} - F_{x3} \tag{B-7}$$

$$m_2\ddot{y}_2 = F_{y2} - F_{y3} \tag{B-8}$$

$$m_2\ddot{z}_2 + m_2 g = F_{z2} - F_{z3} \tag{B-9}$$

$$\begin{aligned} \ddot{\phi}_2 J_{2yy} + \dot{\phi}_1^2\sin\phi_2\cos\phi_2\,\{J_{2zz} - J_{2xx}\} = {} & -F_{z2}k_2\cos\phi_2 + F_{y2}k_2\sin\phi_2\sin\phi_1 \\ & + F_{x2}k_2\sin\phi_2\cos\phi_1 - F_{z3}(a_2-k_2)\cos\phi_2 + F_{y3}(a_2-k_2)\sin\phi_2\sin\phi_1 \\ & + F_{x3}(a_2-k_2)\sin\phi_2\cos\phi_1 + T_2 - T_3 \end{aligned} \tag{B-10}$$

$$\begin{aligned} & \ddot{\phi}_1\,\{J_{2zz}\cos^2\phi_2 + J_{2xx}\sin^2\phi_2\} + 2\dot{\phi}_1\dot{\phi}_2\sin\phi_2\cos\phi_2(J_{2xx} - J_{2zz}) \\ & = F_{x2}k_2\cos\phi_2\sin\phi_1 - F_{y2}k_2\cos\phi_2\cos\phi_1 + F_{x3}(a_2-k_2)\cos\phi_2\sin\phi_1 \\ & \quad - F_{y3}(a_2-k_2)\cos\phi_2\cos\phi_1 + M_2 - M_3 \end{aligned} \tag{B-11}$$

$$m_3\ddot{x}_3 = F_{x3} \tag{B-12}$$

$$m_3\ddot{y}_3 = F_{y3} \tag{B-13}$$

$$m_3\ddot{z}_3 + m_3 g = F_{z3} \tag{B-14}$$

$$\begin{aligned} \ddot{\phi}_3 J_{3yy} + \dot{\phi}_1^2 \sin\phi_3\cos\phi_3 \{J_{3zz} - J_{3xx}\} &= -F_{z3}k_3\cos\phi_3 \\ &\quad + F_{x3}k_3\sin\phi_3\cos\phi_1 + F_{y3}k_3\sin\phi_3\sin\phi_1 + T_3 \end{aligned} \tag{B-15}$$

$$\begin{aligned} \ddot{\phi}_1 \{J_{3zz}\cos^2\phi_3 + J_{3xx}\sin^2\phi_3\} + 2\dot{\phi}_1\dot{\phi}_3\sin\phi_3\cos\phi_3(J_{3xx} - J_{3zz}) \\ = F_{x3}k_3\cos\phi_3\sin\phi_1 - F_{y3}k_3\cos\phi_3\cos\phi_1 + M_3 \end{aligned} \tag{B-16}$$

$$\ddot{x}_1 = -\ddot{\phi}_1 k_1 \sin\phi_1 - \dot{\phi}_1^2 k_1 \cos\phi_1 \tag{B-17}$$

$$\ddot{y}_1 = \ddot{\phi}_1 k_1 \cos\phi_1 - \dot{\phi}_1^2 k_1 \sin\phi_1 \tag{B-18}$$

$$\begin{aligned} \ddot{x}_2 &= -\ddot{\phi}_1\{a_1\sin\phi_1 + k_2\cos\phi_2\sin\phi_1\} - \ddot{\phi}_2 k_2\cos\phi_1\sin\phi_2 \\ &\quad - \dot{\phi}_1^2\{a_1\cos\phi_1 + k_2\cos\phi_1\cos\phi_2\} + 2\dot{\phi}_1\dot{\phi}_2 k_2\sin\phi_1\sin\phi_2 - \dot{\phi}_2^2 k_2\cos\phi_1\cos\phi_2 \end{aligned} \tag{B-19}$$

$$\begin{aligned} \ddot{y}_2 &= \ddot{\phi}_1\{a_1\cos\phi_1 + k_2\cos\phi_2\cos\phi_1\} - \ddot{\phi}_2 k_2\sin\phi_1\sin\phi_2 \\ &\quad - \dot{\phi}_1^2\{a_1\sin\phi_1 + k_2\cos\phi_2\sin\phi_1\} - 2\dot{\phi}_1\dot{\phi}_2 k_2\cos\phi_1\sin\phi_2 \\ &\quad - \dot{\phi}_2^2 k_2\sin\phi_1\cos\phi_2 \end{aligned} \tag{B-20}$$

$$\ddot{z}_2 = \ddot{\phi}_2 k_2\cos\phi_2 - \dot{\phi}_2^2 k_2\sin\phi_2 \tag{B-21}$$

$$\begin{aligned} \ddot{x}_3 &= -\ddot{\phi}_1\{a_1\sin\phi_1 + a_2\cos\phi_2\sin\phi_1 + k_3\cos\phi_3\sin\phi_1\} \\ &\quad - \ddot{\phi}_2 a_2\cos\phi_1\sin\phi_2 - \ddot{\phi}_3 k_3\cos\phi_1\sin\phi_3 - \dot{\phi}_1^2\{a_1\cos\phi_1 + a_2\cos\phi_1\cos\phi_2 \\ &\quad + k_3\cos\phi_1\cos\phi_3\} + 2\dot{\phi}_1\dot{\phi}_2 a_2\sin\phi_1\sin\phi_2 + 2\dot{\phi}_1\dot{\phi}_3 k_3\sin\phi_1\sin\phi_3 \\ &\quad - \dot{\phi}_2^2 a_2\cos\phi_1\cos\phi_2 - \dot{\phi}_3^2 k_3\cos\phi_1\cos\phi_3 \end{aligned} \tag{B-22}$$

$$\begin{aligned}\ddot{y}_3 = {} & \ddot{\phi}_1\{a_1\cos\phi_1 + a_2\cos\phi_1\cos\phi_2 + k_3\cos\phi_1\cos\phi_3\} \\ & - \ddot{\phi}_2 a_2\sin\phi_1\sin\phi_2 - \ddot{\phi}_3 k_3\sin\phi_1\sin\phi_3 \\ & - \dot{\phi}_1^2\{a_1\sin\phi_1 + a_2\cos\phi_2\sin\phi_1 + k_3\cos\phi_3\sin\phi_1\} \\ & - 2\dot{\phi}_1\dot{\phi}_2 a_2\cos\phi_1\sin\phi_2 - 2\dot{\phi}_1\dot{\phi}_3 k_3\cos\phi_1\sin\phi_3 \\ & - \dot{\phi}_2^2 a_2\cos\phi_2\sin\phi_1 - \dot{\phi}_3^2 k_3\cos\phi_3\sin\phi_1 \qquad \text{(B-23)}\end{aligned}$$

$$\ddot{z}_3 = \ddot{\phi}_2 a_2\cos\phi_2 + \ddot{\phi}_3 k_3\cos\phi_3 - \dot{\phi}_2^2 a_2\sin\phi_2 - \dot{\phi}_3^2 k_3\sin\phi_3 \qquad \text{(B-24)}$$

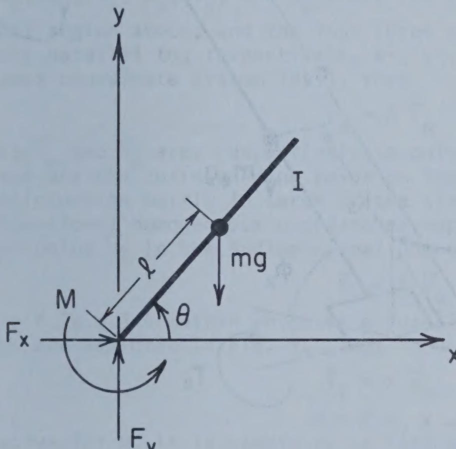


Figure 1: Planar Inverted Pendulum System

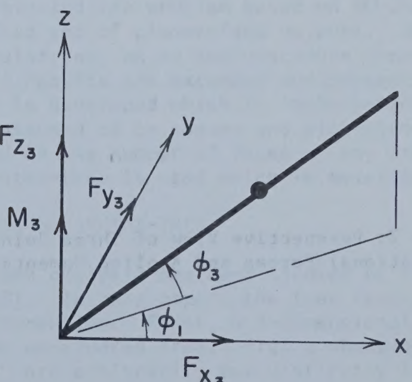
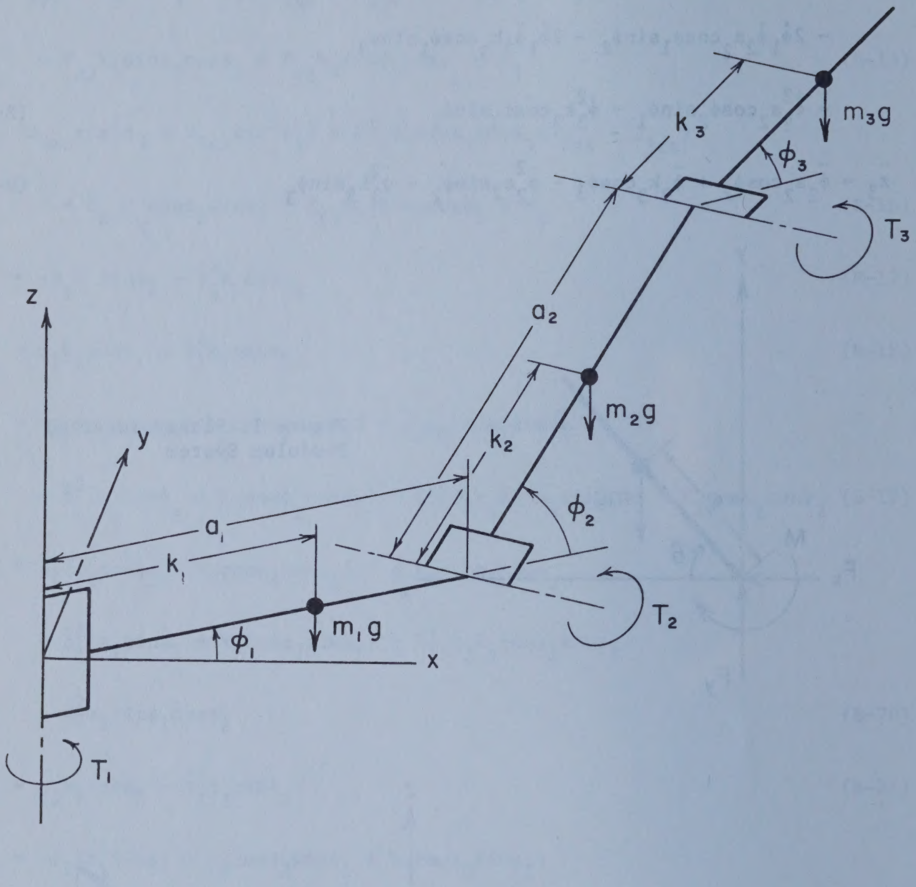


Figure 3: Perspective View of Link 3 Showing Reaction Forces and Moments



k3
m3g
φ3
T3
z
a2
m2g
y
k2
φ2
a1
k1
T2
m1g
φ1
x
T1

Figure 2: Perspective View of Three Joint Manipulator System Showing Gravitational Forces and Applied Moments

THREE DIMENSIONAL PATTERN RECOGNITION FOR THE ROBOTIC SYSTEM

J.Y.S. LUH AND E.S. YAM
School of Electrical Engineering
Purdue University

1. INTRODUCTION

The problem of visual perception for a robotic system is considered. For each job assignment, it is assumed that a fixed set of 3-dimensional objects with known geometrical characteristics is involved. When the vidicon detects one object in the scene, it is required to identify that object, its orientation and its range.

The investigation of perceiving 3-dimensional objects using perspective transformation has been done by Roberts [1], Falk [2], Nitzan [3], and others. Essentially reference models having a fixed orientation and position for each object in the 3-dimensional world-space are chosen so that any arbitrary orientation and position for the same object can be represented by α_x, α_y, α_z, d_x, d_y, and d_z where the first three parameters are rotational angles about, and the last three parameters are translational distances parallel to, respectively, x-, y-, and z-axis. Using the homogeneous coordinate system [4,5], then

$$\overline{V}_o = A\,\overline{V}_m \tag{1}$$

where $\overline{V}_o$ and $\overline{V}_m$ are, respectively, a point on the arbitrarily oriented object and the corresponding point on the reference model; and A is a 4x4 transformation matrix in terms of the six parameters. The point $\overline{V}_o$ in the 4-dimensional homogeneous coordinates maps onto its 3-dimensional, projected image point $\overline{V}_i$ in the 3-dimensional homogeneous coordinates:

$$\overline{V}_i = P\,\overline{V}_o \tag{2}$$

where P is a 3x4 matrix which is a function of local length f. The relations are depicted in Fig. 1. Thus

$$\overline{V}_i = H\,\overline{V}_m \tag{3}$$

where

$$H = P\,A \tag{4}$$

To solve for H, it is necessary to find pairs of corresponding points of the reference model $\overline{V}_m$ and its image $\overline{V}_i$ [1,2]. The point-to-point correspondence problem is not trivial and is the main topic of the following discussion.

Roberts [1] and Falk [2] investigated the problem based on given single line-drawings of a prespecified set of planar-face objects. Using the special features of this particular set, an ad hoc procedure (heuristic programming) was constructed. Their results are extended and presented in this paper. A systematic procedure is developed which is implemented on a digital computer. The objects are assumed to be convex and with planar-faces, but no restriction is imposed on the number of faces or any other special features. The method of syntax-tree is used which is described in the following.

2. CORRESPONDENCE BY SYNTAX-TREE

A. Representation

The syntactic approach to pattern analysis has been studied by Fu [6], Fu and Swain [7], & Fu & Bhargava [8]. In this paper, the tree representations are used for four types of information. First, a 3-dimensional reference model is represented by an un-ordered tree. Fig. 2 shows a "wedge". The "faces" and "vertices" are arbitrarily but distinctly labelled with f's and v's respectively. The corresponding tree has three levels which shows all the faces and the vertices associated with each face. The nodes in each level is not ordered, except that v's of each f are kept in cyclic manner by traversing the vertices of a face (facing the viewer) clockwise. Second, an un-ordered tree represents perspective views. The visible faces of the "wedge" in Fig. 2 are f_4 and f_5 which are depicted in Fig. 3. The five general perspective views and their trees are shown in Fig. 5. Because of un-ordered, one tree represents a number of perspective views of quite different looks, as is evident in Fig. 4. To include more details about the perspective views, an ordered tree may be considered, which uses the concept of line-drawing.

A line-drawing figures is composed of regions r (as opposed to faces in reference model) and nodes n (as opposed to vertices). Its ordered tree representation follows the rule that the "lowest and leftmost" node of the figure is the starting node for a trace, in clockwise direction, of other nodes of a region. 1st junction of the region is then the starting node of the adjacent region. The algorithm of tracing may be modelled as a machine with a push-down stack. A computer program is written for the algorithm (called Algorithm S for simplicity). Fig. 5 shows the input line-drawing of the "wedge" and its ordered tree. To emphasize that the tree is ordered, branches connecting the levels are represented graphically by arrows.

Finally the perspective views of 3-dimensional objects may be represented by ordered trees. As an example, Fig. 4 shows five general perspective view of the "wedge" with f_4 and f_5 visible. Their ordered trees and corresponding label-less trees are also shown. Comparing the latter with the label-less tree in Fig. 5 results an exact match between T_3 and T_i. Thus the correspondence between the vertices of the reference model and the nodes of the input line-drawing is $(v_3, v_2, v_6, v_5, v_4) <=> (n_1, n_2, n_3, n_4, n_5)$.

B. Description of the Method

The schematic diagram of the syntax-tree approach to the correspondence problem is shown in Fig. 6. The method can be divided into two parts.

(a) Off-line training - a "dictionary" containing syntactic descriptions of perspective views (un-ordered trees) and the node-to-vertex correspondences between un-ordered and ordered trees is generated systematically by a computer for the given, fixed set of objects.

(b) On-line perception - a system description S^* (ordered tree) for the line-drawing of some object to be identified is generated and answers of correspondences are obtained by a look-up and match in the dictionary.

3. OFF-LINE TRAINING

Each step in the off-line training process is discussed as follows:

A. Conditions for Simultaneous Visibility of Faces

Given a convex, planar-face object having m faces $F = \{f_1, f_2, \ldots, f_m\}$ with normal unit vectors $N = \{\hat{n}_1, \hat{n}_2, \ldots, \hat{n}_m\}$. Let

$$G(\hat{n}_\ell, a, b, c) = a\,x_\ell + b y_\ell + c z_\ell\,, \quad \ell = 1,2,\ldots,m, \tag{5}$$

be a scalar function where a, b and c are constants to be determined; x_ℓ, y_ℓ and z_ℓ are components of the unit vector $\hat{n}_\ell$. Let

$$ax + by + cz = 0 \tag{6}$$

be a plane through the origin. If there exist a, b and c such that the plane separates U from $\overline{U}$, where U is a subset of F, then all the faces in U are said to be visible. Let V be a subset of N corresponding to U. Then the test condition for visibility is whether one can find a, b and c such that

$$G(\hat{n}_\ell, a, b, c) \begin{cases} > 0, & \text{if } \hat{n}_\ell \in V, \\ \leq 0, & \text{otherwise.} \end{cases}$$

This is equivalent to the problem of linear separability. The solution can be obtained by observing that the solution region in the (a, b, c) space, if it exists, must be in the shape of a convex polyhedral cone diverging from the origin [9]. The "faces" of the cone is represented by

$$G(\hat{n}_\ell, a, b, c) = 0 \tag{7}$$

with a, b and c as variables. An "edge" of the cone is a line joining solutions of a pair of (7). A deterministic algorithm for solving the visibility problem, based on the above observations, is written and called Algorithm V. If the subset U of F exists, it is called "visible subset". The set of all possible "visible subsets" of F is denoted by [F].

B. Tree Grammar for Generating Un-ordered Trees for Perspective Views

With the aid of the visible subset, the collection of un-ordered tree descriptions for all possible perspective views of a given 3-dimensional object can be generated formally by a tree-grammar [6,8]. For illustration, consider the wedge in Fig. 2. The complete collection of the un-ordered-tree descriptions can be generated by the tree-grammar

$$G_W = \{V_N \cup V_T, r, P, S\} \tag{8}$$

where $V_N = \{S, F1, F2, F3, F4, F5\}$, (9)

$V_T = \{f1, f2, f3, f4, f5, v1, v2, v3, v4, v5, v6, \$\}$, (10)

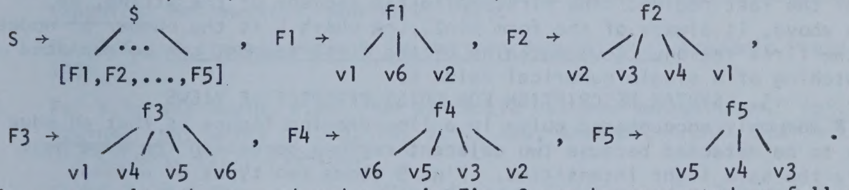
are non-terminal and terminal symbols, respectively; r's are the ranks of the terminal symbols such that

$r(\$) = \{1,2,3\}$, possible numbers of elements in visible subsets (11)

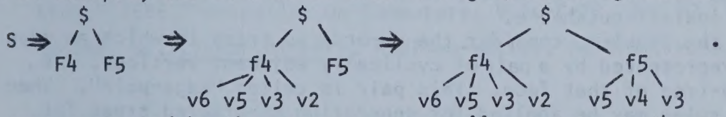
$r(f1) = r(f5) = \{3\}$,
$r(f2) = r(f3) = r(f4) = \{4\}$, } number of vertices of fi, (12)

$r(vi) = \{0\}$, $i = 1,2,\ldots,6$; (13)

S is a start symbol; and P is the set of productions:



As an example, the un-ordered tree in Fig. 3 can be generated as follows:



A tree generated by the grammar necessarily represents the syntax of some observable, perspective views of the object. Since the collection of visible subsets of an object is finite, all un-ordered trees can be enumerated systematically using the grammar.

C. Generation of Ordered Trees from Un-ordered Trees

The following rules are used to obtain a desired ordering:

(a) Any face in the un-ordered tree can be chosen as the first face,

(b) Any vertex of the chosen, first face can be chosen as the first vertex except the one that is the starting vertex of a shared "edge" (pair of cyclically adjacent vertices) with any other face in the tree.

Once the first face and its first vertex are fixed, the ordering of the remaining faces and their vertices are uniquely determined. Algorithm O for generating ordered trees is written based on the above rules. Fig. 5 shows the ordered trees enumerated by the algorithm for the un-ordered tree in Fig. 3.

D. Relabelling and Establishing Correspondences

To provide a link between the face-vertex (f,v) labelling system for 3-dimensional models and the region-node (r,n) labelling system for the line-drawing descriptions, Algorithm C is developed. It accepts an ordered tree of a perspective views in f-v labels as input, and produces an S-description (i.e. ordered tree in regions and modes) and a set of node-to-vertex correspondences. An example is shown in Fig. 7. The output of Algorithm C, called C-description, will be used directly for the correspondence problem during the on-line analysis.

E. Merging and Sorting

Ordered trees representing different line-drawings may have a same C-description (in faces and vertices), which can be merged to form a "compound C-description" containing more than one set of node-to-vertex correspondences. An example is shown in Fig. 8.

F. Dictionary

Using the algorithms discussed above, a collection of all C-descriptions can be systematically generated for any fixed set of 3-dimensional models of any job assignment. Using S-descriptions as key words, a "dictionary" can be compiled which yields the information of node-to-vertex correspondences.

4. REPRESENTATION, STORAGE AND MATCHING OF SYNTAX TREES

Some practical considerations for facilitating the representation, storage and matching of syntax trees (S-descriptions) are discussed below. First, numerical attributes for the tree, such as number of regions, distinct nodes, and nodes in each region can be attached to the tree so that they can be used as indices for the dictionary look-up to save searching time. Second, with delimiting symbols properly added, a syntax can be represented by a one-dimensional string of symbols depicting the last level of the tree. For example, the tree in Fig. 5 can be represented by n1n2n3n4#n4n5n1* where # denotes the end of a region and * denotes the end of the last region. The first delimited segment of the string, as shown above, is always of the form n1n2...nk where k is the number of nodes for the first region. Thus matching of the first segment can be replaced by matching of a single numerical value k.

5. SYNTAX DESCRIPTION FOR NOISY PERSPECTIVE VIEWS

A commonly encountered noise in a line-drawing figure is that an edge fails to be detected because two adjacent regions bordering the edge have nearly the same light intensities. Fig. 9 shows two types of noises: type A - a face is indistinguishable from the background, and type B - two faces are indistinguishable.

To solve the problem, consider the un-ordered trees in which an edge of a face is represented by a pair of cyclically adjacent vertices, v's, within the sub-tree of that face. This pair is called "edge-pair". Then the following rules may be applied for generating un-ordered trees for the noisy perspective views.

(a) Type A noise - a noisy, un-ordered tree can be generated from the correct, un-ordered tree by removing one or more sub-trees for the faces which have unshared "edge-pairs" with other faces in the tree.

(b) Type B noise - a noisy, un-ordered tree can be generated from the correct, un-ordered tree as follows: (i) find a pair of faces fi and fj which have a common "edge-pair" (vm, vn); (ii) rotate the vertices of fi such that vm is the first and vn is the last vertices of the sub-tree fi. If this is impossible, reverse the roles of vm and vn; (iii) rotate the vertices of fj such that vn is the last and vm is the second last in the sub-tree of fj; (iv) remove vm and vn from the sub-tree of fj; and (v) concatenate fi and the remaining sub-tree of fj.

Fig. 9 shows examples of noisy, perspective view with noisy, un-ordered trees. The generation of ordered trees and C-descriptions for the noisy, perspective views from un-ordered trees can be accomplished by the exactly same procedures as used for the noise-less case.

5. CONCLUSIONS

Using the syntax tree approach, a systematic method is developed for solving the correspondence problem between a 2-dimensional line-drawing figure and a set of known, 3-dimensional reference models for convex planar-face objects. The visibility test and tree-grammar are used to algorithmically generate un-ordered trees depicting 2-dimensional perspective views of objects observed at all possible perspective angles. Algorithms are developed to compile a dictionary, which contains the syntax descriptions for all possible perspective views and the desired 2-dimension to 3-dimension correspondence. A push-down automaton is devised for generating syntax descriptions of unknown input line-drawing figures. Consequently the on-line solution to the correspondence problem involves a simple dictionary look-up. A method of algorithmically generating syntax descriptions for noisy line-drawing figure is also presented.

6. REFERENCES

1. Roberts, L.G., "Machine Perception of Three-dimensional Solids," In Optical and Electro-optical Information Processing, Tippett et al. (Eds), MIT Press, Cambridge, Mass., 1965, 159-197.

2. Falk. G., "Interpretation of Imperfect Line Data as a Three-Dimensional Scene," Artificial Intellegence, 3 (1972), 101-144.

3. Nitzan, D., "Object Recognition in Multisensory Scene Analysis," Technical Note 83, Artificial Intelligence Center, Stanford Research Institute, Dec. 1973.

4. Duda, R.O. and Hart, P.E., Pattern Recognition and Scene Analysis, Wiley, New York, N.Y., 1973.

5. Newman, W.M. and Sproull, R.F., Principles of Interactive Computer Graphics, McGraw-Hill, New York, 1973.

6. Fu, K.S., Syntactic Methods in Pattern Recognition, Academic Press, 1974.

7. Fu, K.S. and Swain, P.H., "On Syntactic Pattern Recognition," Software Engineering, COINS III, 2, Ed. J. T. Tou, Academic Press, New York and London, 1971, 155-182.

8. Fu, K.S. and Bhargava, B., "Tree Systems for Syntactic Pattern Recognition," IEEE Transaction on Computers, Vol. C-22, No. 12, Dec. 1973.

9. Hadley, G., Linear Algebra, Addison-Wesley, Reading, Mass., 1961.

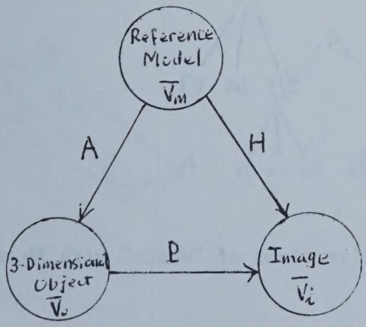


Fig. 1. Mapping of Image from Reference Model and 3-Dimensional Object

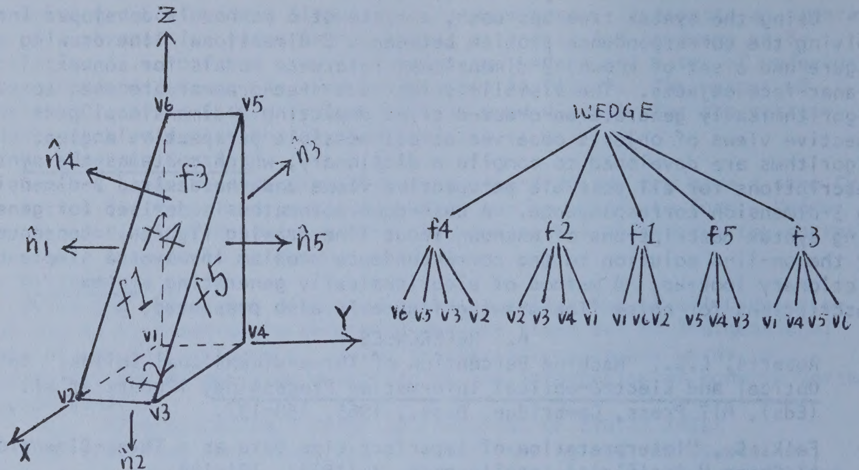

Z
V6
V5
n̂4
f3
n̂3
n̂1
f4
n̂5
f1
f5
V1
V4
Y
V2
V3
X
n̂2
2
WEDGE
f4
f2
f1
f5
f3
V6 V5 V3 V2
V2 V3 V4 V1
V1 V6 V2
V5 V4 V3
V1 V4 V5 V6

Fig. 2. Input 3-d Descriptive Model for the "Wedge".

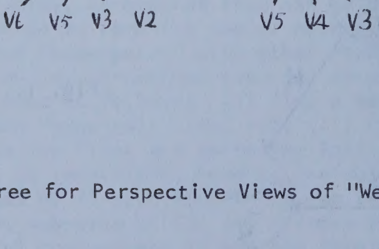

Perspective View of "Wedge"
f4
f5
V6 V5 V3 V2
V5 V4 V3

Fig. 3. Un-ordered Tree for Perspective Views of "Wedge" with f4 and f5 Visible.

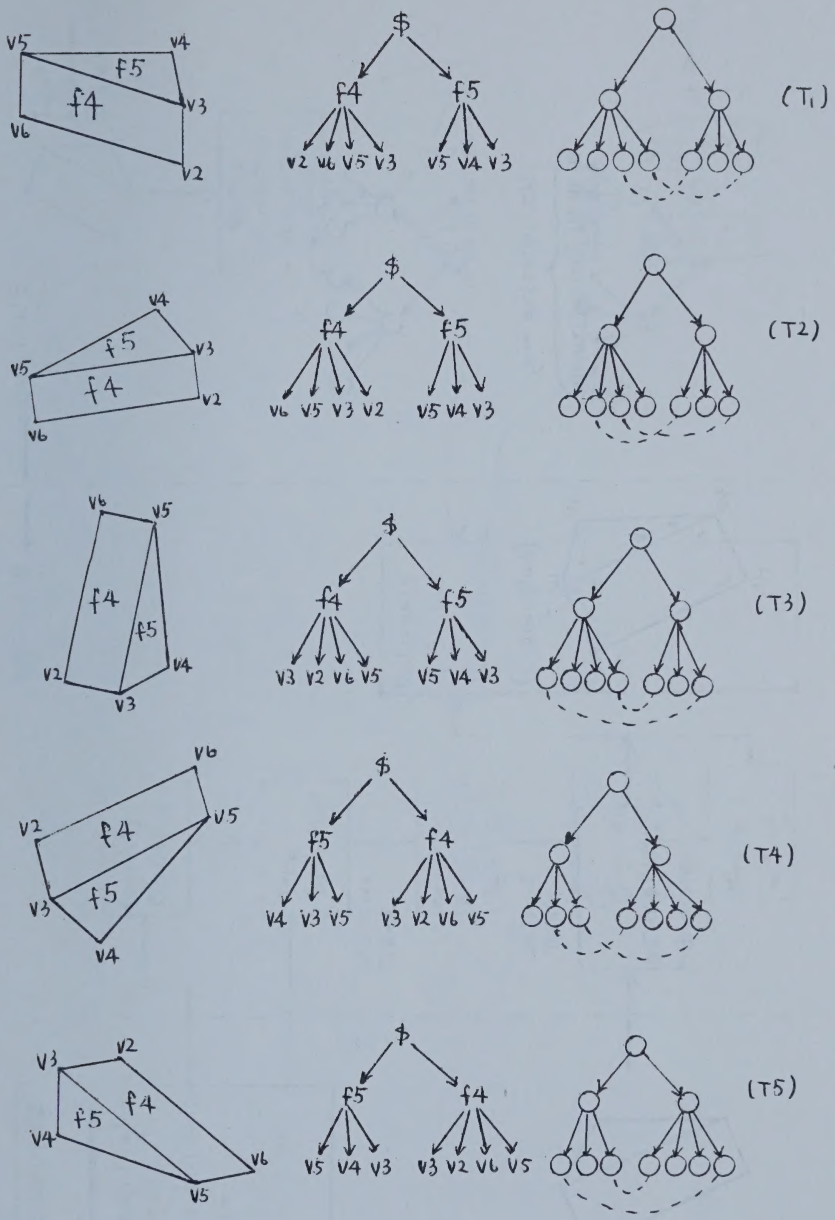

V5 V4 V3 V6 V2 f5 f4
$ f4 f5
V2 V6 V5 V3 V5 V4 V3
(T1)
V4 V3 V5 V2 V6 f5 f4
$ f4 f5
V6 V5 V3 V2 V5 V4 V3
(T2)
V6 V5 f4 f5 V2 V4 V3
$ f4 f5
V3 V2 V6 V5 V5 V4 V3
(T3)
V6 V5 V2 V3 V4 f4 f5
$ f5 f4
V4 V3 V5 V3 V2 V6 V5
(T4)
V3 V2 V4 V6 V5 f5 f4
$ f5 f4
V5 V4 V3 V3 V2 V6 V5
(T5)

Fig. 4. Ordered Trees and Perspective Views.

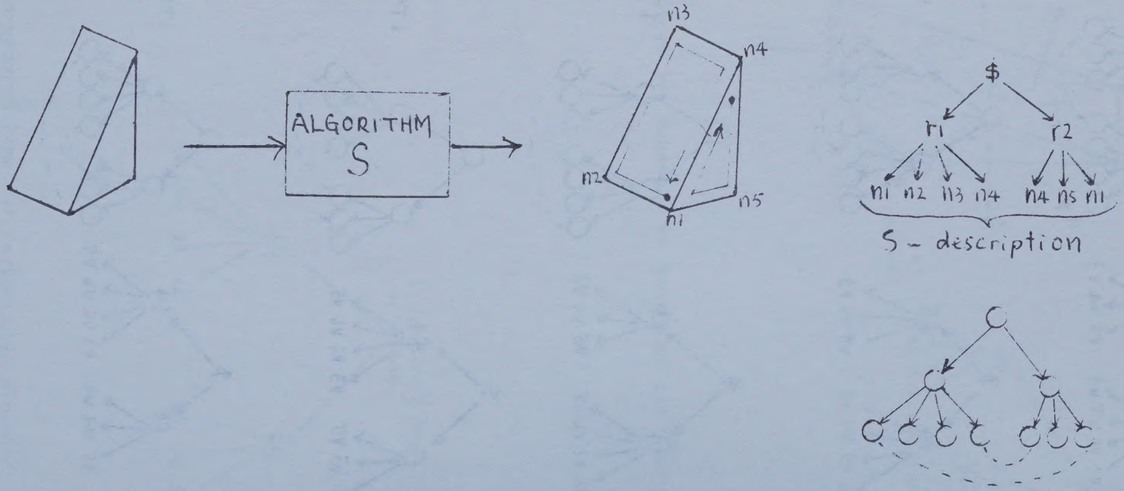

ALGORITHM
S
n3
n4
n2
n1
n5
$
r1
r2
n1 n2 n3 n4
n4 n5 n1
S - description

Fig. 5. Algorithm-S.

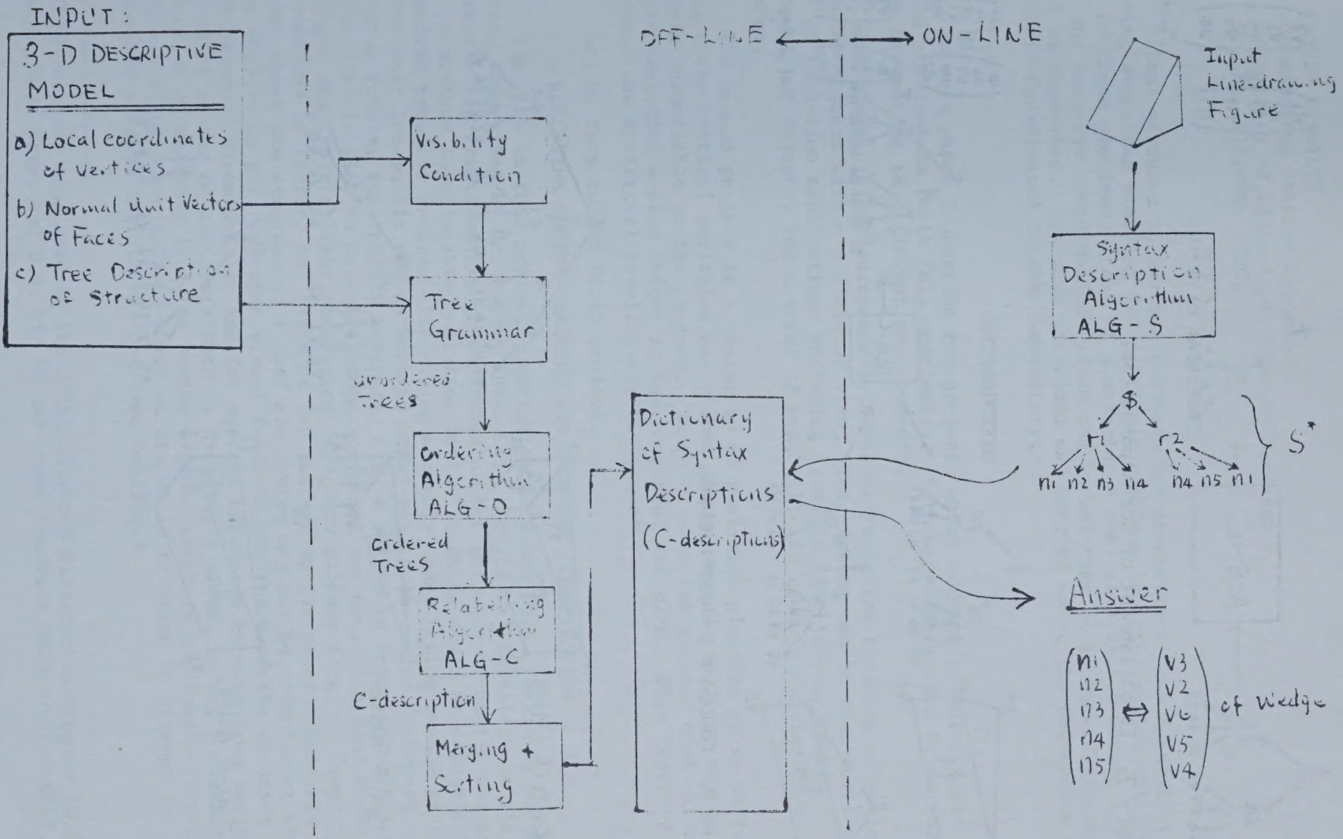

INPUT:
3-D DESCRIPTIVE MODEL
a) Local coordinates of vertices
b) Normal unit vectors of Faces
c) Tree Description of Structure
Visibility Condition
Tree Grammar
unordered Trees
Ordering Algorithm ALG-O
ordered Trees
Relabelling Algorithm ALG-C
C-description
Merging + Sorting
OFF-LINE
ON-LINE
Dictionary of Syntax Descriptions (C-descriptions)
Input Line-drawing Figure
Syntax Description Algorithm ALG-S
$
r1
r2
n1 n2 n3 n4
n4 n5 n1
S*
Answer
n1 n2 n3 n4 n5
V3 V2 V6 V5 V4
of Wedge

Fig. 6. Schematic Diagram for the Correspondence Problem.

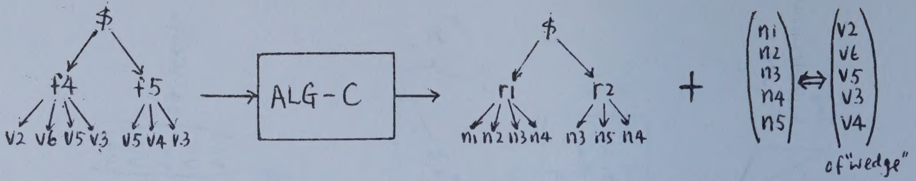

$
f4
f5
V2 V6 V5 V3
V5 V4 V3
ALG-C
$
r1
r2
n1 n2 n3 n4
n3 n5 n4
+
n1 n2 n3 n4 n5
⇔
V2 V6 V5 V3 V4
of "wedge"

Figure 7. Establishment of Correspondences.

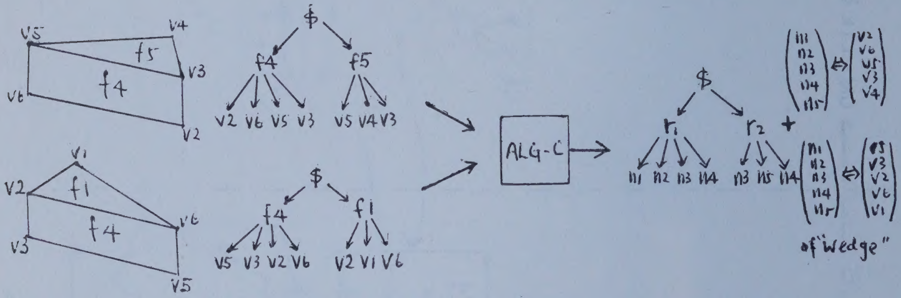

V5
V4
f5
V3
f4
V6
V2
V1
V2
f1
f4
V6
V3
V5
$
f4
f5
V2 V6 V5 V3
V5 V4 V3
$
f4
f1
V5 V3 V2 V6
V2 V1 V6
ALG-C
$
r1
r2
n1 n2 n3 n4
n3 n5 n4
+
n1 n2 n3 n4 n5
⇔
V2 V6 V5 V3 V4
n1 n2 n3 n4 n5
⇔
V5 V3 V2 V6 V1
of "wedge"

Figure 8. Compound C-descriptions.

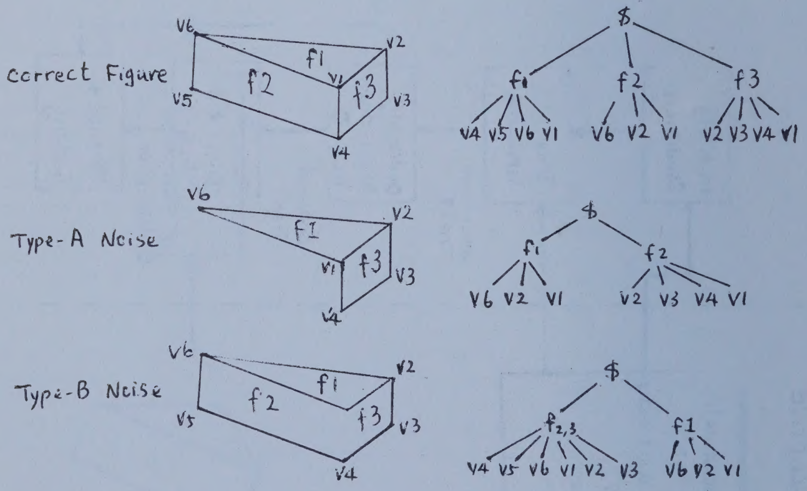

Correct Figure
V6
V2
f1
f2
V1
f3
V5
V3
V4
$
f1
f2
f3
V4 V5 V6 V1
V6 V2 V1
V2 V3 V4 V1
Type-A Noise
V6
V2
f1
V1
f3
V3
V4
$
f1
f2
V6 V2 V1
V2 V3 V4 V1
Type-B Noise
V6
V2
f1
f2
f3
V5
V3
V4
$
f2,3
f1
V4 V5 V6 V1 V2 V3
V6 V2 V1

Fig. 9. Two Types of Noisy Perspective View.

NEW PROGRESS IN ROBOTICS*

DAVID L. WALTZ
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

ABSTRACT

Recent advanced automation research has demonstrated the feasibility of solving some long standing and difficult problems in artificial intelligence. Two of these problems, those of generating a real-time vision motion tracking and hand-eye coordination system and a real-time decision aid system, will be described. The work described was carried on at the University of Illinois Coordinated Science Laboratory.

INTRODUCTION

In this paper I describe two projects which attempt to solve real-world problems in their full complexity. Both systems attempt to do real-time processing in different environments.

The first project successfully achieves real-time tracking of motion of several objects within a field of view. This system can be used for a number of real-time applications including assembly line tasks, precise assembly of objects, and control of pan, tilt and zoom facilities of a camera.

The second project is a system to do real-time planning in an environment where critical variables may either change unpredictably or may be simply unavailable to the program. In such cases the program must often begin executing a plan before it has completed the plan. This requires special new artificial intelligence techniques.

Let us turn to the first project.

REAL-TIME MOTION TRACKING AND HAND-EYE COORDINATION

The work in this project constitutes a recent Ph.D. Thesis by Vincent Jones, supervised by Dr. Robert Chien [1]. Artificial intelligence vision research has traditionally concentrated on the analysis of static scenes. It is obvious that dynamic real-time motion tracking would be valuable for handling problems dealing with moving assembly lines having random parts placement; for mobile robots such as under sea or deep mine explorers or Martian rovers; for traffic monitors, e.g. a system to input information about traffic flow to a central computer which can then plan lane marker changes and traffic light sequencing; for surveillance (e.g. of "empty" warehouses or stores) for fires or for intruders; for precise assembly tasks where the accuracy of a hand arm system may not be adequate or where the ability to get real-time visual feedback can increase the accuracy of the assembly tremendously; and for systems which can recover from various kinds of errors or accidents such as dropping a piece, having a missing piece on an assembly line, or having a piece which is improperly formed. (This will help avoid cases where an arm would continue to attempt to assemble a piece that it was not even holding.)

---

*This work was supported by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-72-C-0259.

There is no easy way to modify traditional visual recognition programs to do this sort of tracking. Traditional recognition programs are simply too slow; even the fastest programs require time periods on the order of seconds to analyze a scene. Clearly within this time a moving object can completely disappear from the field of view. It is obvious that the slowness of these recognition programs is not a matter of bad programming or lack of ideas. Each of the two interlaced frames of a standard size TV picture contains 60,000 points; to access every point in the picture without even doing any further processing requires about 60 milliseconds on current generation computers. The solution we have used has been to separate the tasks of tracking and recognition. The traditional artificial intelligence scene recognizer operates on a full frame at its normal repetition rate (on the order of 5 seconds per picture). Such a recognizer operates on a full frame and is included as part of our system (see Fig. 1).

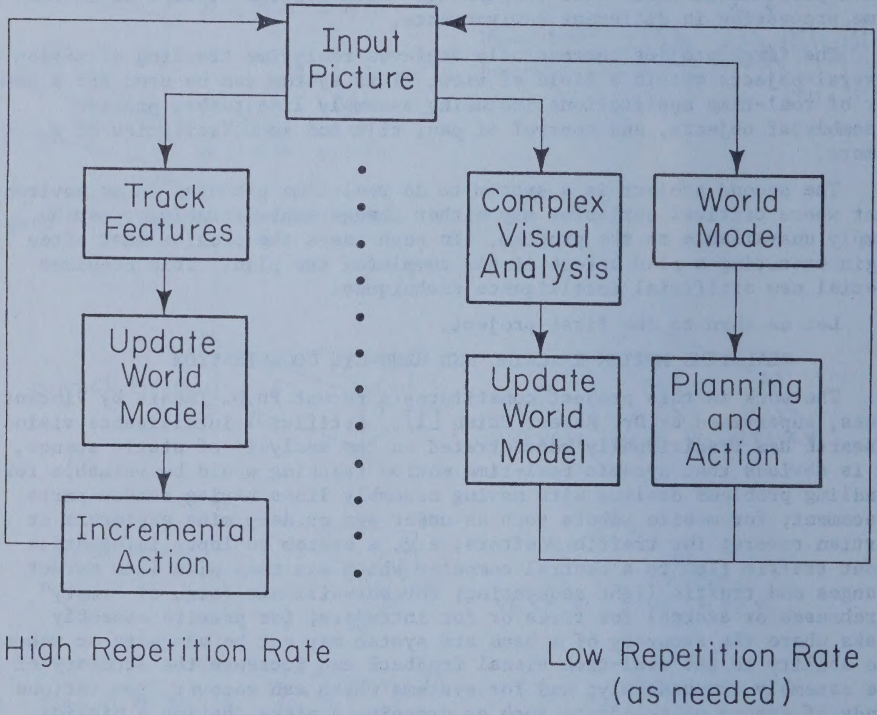

Input Picture
Track Features
Update World Model
Incremental Action
Complex Visual Analysis
World Model
Update World Model
Planning and Action
High Repetition Rate
Low Repetition Rate (as needed)

Figure 1. Tracking (continuous) vision approach.

Our tracker operates on a window or windows of the scene which are 36 x 36 pixels in size. The tracker operates at a high repetition rate of 10 frames per second or more.

The tracker does no recognition whatsoever, but merely keeps track of simple features like edges, vertices, prominences, or dark areas on an object. A supervisor attached to the TV camera allots time to the tracker and recognizer and controls the data flow to each of these modules.

The value of this approach has been shown in a number of demonstrations:

(1) first, the system can control the pan/tilt head and zoom features of our TV camera. In a sample demonstration, a human operator can hold a piece of paper with a dark spot in front of the camera. Once the tracker has located the dark spot, the person can then move the piece of paper around at a rapid rate in front of the camera and the pan/tilt head will track the spot, keeping it centered in the computer's field of view. Examples of the utility of zoom control will be given later.

(2) In another demonstration, the system can track several objects at one time. Several people stand outside the computer's work space holding strings attached to objects in the computer's work space. When the computer has located the objects, the people can move the objects around simultaneously; the computer tracks all the objects and prints the tracks of the objects as they move on a TV monitor. The only difficulty in this demonstration occurs when two objects touch or come so close as to appear to become one object. The computer may not be able to recognize which object is which when they separate again. In this demonstration, the computer can track up to seven or eight objects simultaneously.

(3) A traffic monitor. In this demonstration, the TV camera is pointed outside the window of the laboratory at a nearby intersection. As cars, bicycles, trucks, buses and pedestrians appear in the intersection, the computer tracks them, does an elementary recognition procedure on each item and displays the results of its analysis in a symbolic form on a TV monitor (e.g. it shows a stylized car moving on a street). This demonstration was never completely successful, not because of a failure of the vision scheme, but because of a hardware difficulty. Our TV camera has a special low bloom property (bloom is the characteristic of vidicon cameras which causes very bright areas to appear larger than they are). Our camera was specially designed with low bloom characteristics, i.e., on our camera, very bright regions appear to be approximately their real size. Unfortunately, in reducing bloom, we have also increased the persistence of images in the camera, and this leads to blurring of objects that move too fast through the field of view. Consequently, recognition procedures have not been able to reliably track and recognize items in the traffic demonstration.

The two following demonstrations have been filmed and are available in a 16 mm movie from the Coordinated Science Laboratory.

(4) Using real-time visual feedback, our computer hand is able to stack objects (in this case blocks) very precisely, using a mirror placed near the objects to be stacked to give the equivalent of a binocular view of the scene. In this demonstration, the hand and eye are not calibrated, but the hand does have a distinctive mark that enables the TV camera to locate it. The demonstration begins with the hand holding a block which it is to place on top of a base block located on its work surface. Once the hand and the base block have been located, the hand moves toward the base block until both the base block and the block to be stacked can be seen in the mirror. The hand then lowers the block it is holding toward the base block, aligning edges both in its normal view and in the mirror view as it descends. The mirror and base block can be moved around as the hand descends and the

demonstration will still succeed since the program operates simply by tracking the corners of the top block and the corners of the base block. In this demonstration, the recognizer is not used at all. Despite the fact that our hand is not very accurate and the fact that the system is not calibrated, extremely precise stacking has been demonstrated.

(5) The final demonstration involves inserting a pin in a moving hole. In this demonstration, a human operator moves a small object with a hole (see Fig. 2) around on the work surface. As the object moves, the program takes a full frame picture and passes it to the recognizer, meanwhile tracking the object with the hole in it. The recognizer attempts to identify the shape of the hole and once it has done so, the hand selects an appropriate pin from a rack on the work surface (the different shapes are square, round, triangular, etc.). The hand then attempts to insert the pin in the hole while the human operator continues to move the object around on the table. As the pin approaches the hole, the computer zooms in for a closer, more precise view. The only constraints on this demonstration are that the human operator must move the block slower than the speed at which the hand can move and that the computer must be calibrated for the work surface height. In this case, the movie demonstration is not as impressive as it might be because the hand can move at a rate of only 1 inch a second. However, the vision system is capable of handling motion much faster than this and when our new Scheinman arm [8] is available, a considerably more impressive demonstration will be possible.

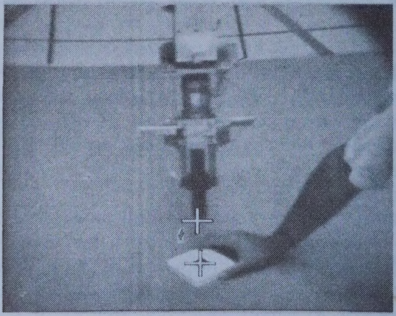


Figure 2. TV monitor view of the INSERTION demonstration.

POTENTIAL VALUE FOR APPLICATIONS

Such a system would also be valuable for a moving robot. In traditional moving robots such as Shakey at SRI [6] each time the robot moves, it must completely recompute the scene surrounding it. In its normal mode, it will then plan another traversal and after accomplishing this motion, completely recompute its environment once again. Clearly with a real-time motion system such as the one described in this paper, a robot could continually monitor its surroundings and would have no need to generate a correspondence between scenes from the ground up.

Another traditional problem which can be solved well by this system is the problem of resolution in the imaging device. Most artificial intelligence systems have operated either with a vidicon or with a vidissector. Theoretically, the vidissector has extremely high resolution (up to 1000 x 1000) but because it is a random access device, the rate at which points can be read is extremely slow in order to reduce noise to an acceptable level. On the other hand, a vidicon can operate very rapidly but has limited resolution, usually around 250 x 250 points. The problem in using a zoom lens is that even when an object to be examined closely is centered in the field of view, when the camera zooms in on the object (especially if the object is moving) the object may disappear off the side of the field of view. Therefore in the past, careful calibration (see Sobel [9]) has been necessary to make sure that objects remain in the field of view; if the objects are moving, past systems simply have no chance of success. With our system it is possible to track a feature as the camera zooms in for a closer look, keeping the feature approximately centered in the field of view and eliminating the problem of losing track of the object.

PLANNING IN AN UNCERTAIN ENVIRONMENT

Two years ago we began work on a computer aided decision making (CADM) system for automatic aid to pilots in failure detection and correction in an aircraft cockpit environment. The task of the decision maker was to plan a course of action once a failure or difficulty had been identified. Traditional planners, e.g. PLANNER (Hewitt [5]), GPS (Ernst and Newell [3]), STRIPS (Fikes and Nilsson [4]), assume a fixed planning sequence:
1) generate a plan;
2) execute the plan.

There were severe problems with such schemes:
(1) there is not always enough time within a cockpit or real-time environment to complete and verify a plan before execution can begin, and a delay in execution in such an environment can be disastrous;
(2) a complete plan may no longer be appropriate when it is executed; the environment in an aircraft can change very rapidly and so certain critical variables may no longer have their original values at execution time;
(3) the values of some critical variables may be inaccessible, either totally (in some contexts) or until part of the plan has already been executed; if this happens, it is simply impossible to generate and verify a complete plan.
In cases like these, older planners either fail completely or must generate totally new plans at each point where a variable changes or is found to be inaccessible.

Our solution to this problem has been to generate a system which needs only to determine that a plan can succeed with some confidence level before it can begin executing the plan; it completes the plan in parallel with the plan execution. In this system, values are assigned to each procedure which can be part of the plan to indicate:
(1) the level of importance of the procedure and
(2) the expected difficulty of achieving this procedure.
(These values are now input by a programmer; in the future these values could be based on the program's experience.)

These values are used two ways. The first is to form "important" parts of the plan first, and fill in the details later, in a manner somewhat similar to that used by Sacerdoti [7]. The values are also used to assign a relative likelihood of success to different plans and plan fragments before filling in the details of the plan. Once a plan with an acceptable chance of success has been formed, execution can begin. For critical real-time situations, the level of acceptability can be lowered whereas if more time is available the level of acceptability can be made very high. Thus the system is able to adjust its planning and execution ability to the context in which the problem solver is operating.

The system sets up demons for each goal which is not yet part of the verified plan. (A "demon" is a pattern matching procedure which watches for its pattern to be satisfied; if its pattern is ever satisfied, the demon interrupts the normal procedure of the program, performs some operation or procedure of its own and then returns control to the original program). If goals for which the demons are designed are ever satisfied unexpectedly by either side effects of other procedures or by dynamic changes in the environment, then the system has found a "short-cut." Thus the system may actually realize a considerable savings in time and planning resources over a system which forms a complete plan before beginning execution.

A demonstration has been given of the operation of this system. A portion of an aircraft system has been simulated (see Fig. 3). In this

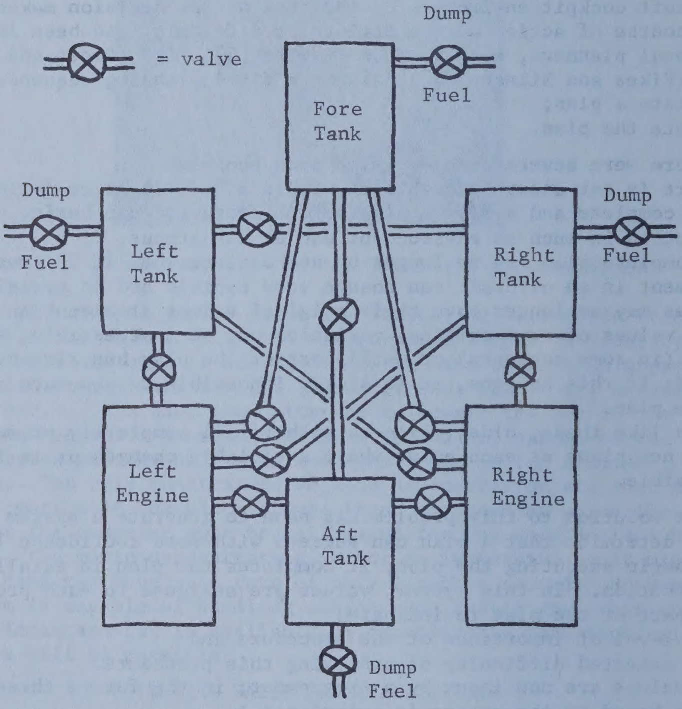


Figure 3. A simulated aircraft fuel system for use with CADM program.

example a portion of a plane's fuel and engine system is simulated. This simulation operates in conjunction with a complete aircraft simulator which provides a visual display for the pilot on a GRT. A pilot or operator of the system must perform a tracking task while attempting to correct failures in the fuel system of the aircraft. In contrast, when the CADM system is operating, the pilot need only do the tracking task while the CADM system handles the correction of failures in the fuel system. Typical failures are flame-out of an engine, failure of a valve, the need to adjust CG (center of gravity) of the aircraft, and so on.

This planning system is described in more detail in Chien and Weissman [2].

CONCLUSION

The ideas underlying two systems to perform real-time operation have been described. We feel that these systems are representative of a new trend in artificial intelligence away from "toy problems" and toward the solution of real world problems in their full complexity. This research could not have been done without a body of basic research on which to draw. However, it is now clear that the time is ripe for real world, real-time applications of artificial intelligence. Such research is not only valuable for its practical applications; it can yield insight into problems of a general nature which might never be discovered in any other way.

REFERENCES

[1] R. T. Chien and V. C. Jones, "Acquisition of Moving Objects and Hand-Eye Coordination," Adv. Papers Fourth Int'l. Joint Conf. on A. I., Tbilisi, Georgia, USSR, Sept. 1975, pp. 737-741.

[2] R. T. Chien and S. Weissman, "Planning and Execution in Incompletely Specified Environments," Adv. Papers Fourth Int'l Joint Conf. on A.I., Tbilisi, Georgia, USSR, Sept. 1975, pp. 737-741.

[3] G. W. Ernst and A. Newell, GPS: A Case Study in Generality and Problem Solving, Academic Press, New York, 1969.

[4] R. E. Fikes and N. J. Nilsson, "STRIPS: A New Approach to the Application of Theorem Proving in Problem Solving," Artificial Intelligence, Vol. 2, pp. 189-208, 1971.

[5] C. Hewitt, "Description and Theoretical Analysis (Using Schemata) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot," Ph.D. thesis, Dept. of Math., M.I.T., Cambridge, MA, 1971.

[6] B. Raphael et al., "Research and Applications -- Artificial Intelligence," Stanford Research Institute Final Report, Contract NASA-2164, National Aeronautics and Space Administration, Dec. 1971.

[7] E. D. Sacerdoti, "Planning in a Hierarchy of Abstraction Spaces," Adv. Papers Third Int'l Joint Conf. on A. I., Stanford Univ., Stanford, CA, Aug. 1973, pp. 412-422.

[8] V. D. Scheinman, "Design of a Computer-Controlled Manipulator," Memo AIM-92, Stanford Artificial Intelligence Lab., Stanford Univ., Stanford, CA, June 1969.

[9] I. Sobel, "On Calibrating Computer Controlled Cameras for Perceiving 3-D Scenes," Adv. Papers Third Int'l Joint Conf. on A. I., Stanford Univ., Stanford, CA, Aug. 1973, pp. 648-657.

USING A MICROCOMPUTER TO CONTROL A SMALL ROBOTIC CART

RICHARD A. NORTHOUSE
University of Wisconsin-Milwaukee
Milwaukee, Wisconsin 53201

ABSTRACT

This paper reports on preliminary investigations into the research and development of a small telerobotic cart. The cart has an onboard manipulator for handling small objects, and onboard sensors to monitor its own status. This cart is connected to a background computer via two unidirectional (full duplex) telemetry links.

The cart is controlled by a mini-computer, and has a small microprocessor onboard the cart for simple intelligence. The mini-computer will process commands issued by an operator and also furnish this operator with a visual display. This combination of teleoperator and robot we have called telerobot.

The engineering significance of this research is to: (a) unite a teleoperator system with onboard intelligence (this onboard intelligence is in the form of a small microprocessor), (b) employ a mini-computer to assist the operator in his decisions (this mini-computer will permit the operator to use an English-like language to control the telerobot), (c) the use of small computers such as the mini- and microprocessors could significantly reduce the purchase and operating costs of teleoperators. The microprocessor also provides a means for easy retraining of the telerobot.

INTRODUCTION

Teleoperator and robot research and development are at the present time in a state of evolution. Applications for both concepts extend from outer space and underwater exploration to a full range of terrestrial and industrial implementations.

The basic teleoperator concept is the amplification, extension and communication of man's dexterity across barriers such as hostile environments, task difficulty and distance. Implicit, of course, in the teleoperator system is the human operator who makes decisions and sends commands to the teleoperator based on feedback information provided by sensors mounted on or near the teleoperator.

The robot concept, although much popularized as a Frankenstein-like creature, is better defined as an autonomous problem solver and task executor. In terms of teleoperators, robots can be partially described as a teleoperator system with the human operator replaced with some artificial intelligence.

Research has been recently directed at augmenting teleoperators with computer processing because of task complexity and transmission delay over long distances. Some of the ways computers would be used with teleoperators is in aiding the operator by means of sensor data processing, stored task programs, forecasting consequences of particular actions, and storing results of good and bad moves for later use. Also, computer control would be implemented on the teleoperator proper to provide a set of automatic reflexes for insuring survival of the teleoperator. This is important when there is a great distance between the teleoperator and the operator, and consequently, transmission delays would prevent the operator from reacting quickly (or if the transmission link were temporarily broken).

Research in robots is generally dispersed among several related and necessary fields such as artificial intelligence, pattern recognition, automatic control, adaptive control, remote control, automata, and computer science. Present robots do not, however, exhibit a sufficient level of autonomy. A general purpose, completely independent, sophisticated robot does not appear to be immediately realizable.

The limitations of the teleoperator concept over long distances, the shortcomings of the human operator in a complex task, high stress situations, and the lack of sophistication in state of the art robots for complex problem environments suggests a marriage between the concepts of teleoperators and robots. This combination would provide the necessary requirements of intelligence and response time for effecting a complex task in a distant environment. This combination of teleoperator concepts with those of robotics will be termed a telerobot.

This research develops a telerobot with a simple configuration of hardware consisting of a motorized cart equipped with a simple manipulator, an onboard micro-computer, a diode matrix camera,a background mini-computer, an operator interface (TV monitors, CRT, and keyboard) and communication equipment. Since sophisticated, hardware-oriented teleoperators presently exist, software is emphasized in this work, rather than expensive and difficult to maintain hardware. Also, to develop general problem solving and task execution capabilities, a software approach is felt to be essential.

LITERATURE SURVEY

The fields of telerobots can be broken down into the following classes:

1. teleoperators
2. visual pattern recognition
3. navigation techniques and algorithms for mobile automatons
4. problem-solving techniques
5. fixed mechanical systems
6. mobile systems.

Teleoperators

A full bibliography, history and write-up of teleoperators exist in very readable form in Teleoperators and Human Augmentation (1967) (An AEC - NASA Technological Survey). Other information can be found in Handbook on Telemetry and Remote Control (Gruenberg - 1967).

The use of computers in limited decision making and command generation has been termed "supervisory control" by Prof. Thomas B. Sheridan (1968) and his co-workers at Massachusetts Institute of Technology.

Visual Pattern Recognition

This category has received much attention since the applications for this type of system extend beyond the field of Robotics and Teleoperators. The book by Rosenfeld (1969) is fairly comprehensive text covering the subject of visual pattern recognition by computer. The paper by Roberts (1962) is the basis for many of the visual systems used in existing robots. Pingle (1968) explains how an edge-tracing algorithm can be implemented to find objects in a scene. The basic input to his system is a vidicon camera. As in almost all existing systems, the camera scan pattern is sampled at discrete points and the intensity levels are digitized. This information is then sent to a computer where the scene analysis takes place. Additional edge-detection methods are outlined by Yachida and Tsuji (1971), Guzman (1968), Forsen (1968) and Kelly (1970). Pflatz, et al (1968) talk about some of the picture processing computer programs which have been developed. Fargo and Northouse (1974) illustrate how a mini-computer can be used for real time character recognition. Although quite a lot of work has been done in the field of visual pattern recognition, much still remains to be done to speed up the computer processing time and reduce the storage requirements.

Navigation Algorithms

In the area of navigation, Nilsson (1967) has developed a system based on a maze-solving algorithm developed by Moore (1959). The system is flexible in that the robot finds and remembers the number and locations of obstacles. The shortest path to the goal is found, although in some instances the amount of computations required may be large. Keckler and Larson (1970) determine the best path to the goal via probabilities of obstacle position occurrence, hence, the term "partially unknown". The Jet Propulsion Laboratory (JPL TR 32-1288) has developed an algorithm for reaching a target by following terrain contours. The algorithm does not necessarily find the shortest path. The trade-off is increased simplicity and speed of execution. Kocourek and Northouse (1973) have developed some mini-computer implementable navigation algorithms. They sacrifice elegance for speed and a minimum amount of computer resources.

Problem Solving Techniques

Problem solving behavior has been investigated from the viewpoint of processing natural language conversation and from the viewpoint of constructing plausible courses of action. Common to all serious projects has been the incorporation of theorem-proving techniques as the deduction basis for goal-directed activity. Question answering systems which utilize theorem-proving have been surveyed by Greene and Raphael (1968). In these cases, correctly answering questions is tantamount to correctly implementing problem solutions. Program construction and verification research has not always been related to Robotics (see, e.g., Wegner (1972)). Most closely allied with robotics projects are the synthesis methods described by Manna and Waldinger (1971). In addition, behavior based on incompletely specified commands has been given initial treatment by Chang (1972). Documentation of major problems in related areas can be found in Minsky (1969).

Fixed Mechanical Systems

A number of robot systems have been built or are proposed. The first was a fixed-arm system with no visual or audio input and was developed by

Ernst (1962). Its contact with its environment consisted of microswitches and photodiodes. Later systems incorporated some of the visual techniques described above, (Pingle, et al., 1968, Ejire, et al., 1972). The system described by McCarthy, et al., (1968), uses a visual input channel as well as an audio channel.

Mobile Systems

Comparatively little work has been done so far in the area of mobile robots. Most of the existing robots are fixed devices with only movable arms or hands (such as Kilmer and Sutro, 1969). The work by Northouse, et al. (1972) and Leverance, Kocourek and Northouse (1973) describes work done on a small simple mobile system called O'Raily. Another mobile robot was developed at the University of California-Berkeley. This fellow was called Jason (Smith and Coles, 1973). Hager and Northouse (1975a) recently modified O'Raily and call it Ralph.

Some of the problems that need to be solved include:

1. the development of fast, efficient algorithms for visual object recognition
2. the above algorithms must be efficient with their use of computer memory
3. power requirements for locomotion and "thinking" must be reduced
4. development of fast, efficient algorithms for navigation and object manipulation
5. the interface with problem-solving programs.

Some criteria for workable systems are given by Meitus (1969), Kilmer and Sutro (1969), and Hager and Northouse (1975a and 1975b).

RALPH, THE TELEROBOT

Our telerobot is basically the motorized cart with a mechanical manipulator as described by Hager and Northouse (1975a) and a matrix diode camera for providing visual feedback. Control of the telerobot is done on three levels which are: (a) a human operator, (b) a background minicomputer, and (c) an onboard microprocessor. Figure 1 illustrates the system.

The mechanical system is hardware-simple, with the sophistication emphasized in the software. The software will be developed in terms of maximizing autonomy of the teleoperator and providing an interfacing for the operator and supportive resources.

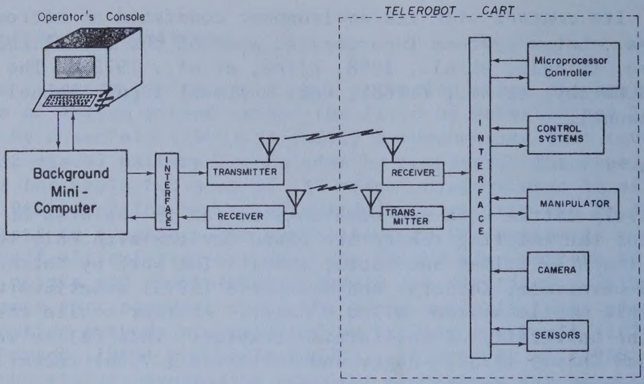

Operator's Console
TELEROBOT CART
Background Mini-Computer
INTERFACE
TRANSMITTER
RECEIVER
RECEIVER
TRANS-MITTER
INTERFACE
Microprocessor Controller
CONTROL SYSTEMS
MANIPULATOR
CAMERA
SENSORS

Figure 1. Block Diagram of the Telerobot System.

TELEROBOT SUBSYSTEMS

Robot Cart

The actuator is a motorized cart with a manipulator-claw mounted on it. The cart is 18"inches in diameter and 30" high. It is powered with two 1/4 horsepower permanent magnetic d.c. motors. A single 12 volt storage battery provides the cart with electrical power. The cart is described in detail by Hager and Northouse (1975a) and is illustrated by Figure 2. The manipulator or mechanical hand is currently in final development stages and will be of simple design consisting of six degrees of freedom. Although the hardware is somewhat unsophisticated, the interfacing and associated software will be made general so as to allow future additions of sophisticated equipment if it is felt necessary. Figure 3 illustrates the manipulator.

Sensors

The sensor subsystem is the sentient portion of the telerobot in that it provides the operator and the automatic reflexes with the data from the environment. There are four areas that require sensor information:

1. vehicle navigation
2. task status
3. telerobot status
4. device status.

1) Vehicle navigation requires positional information (such as range and direction), obstacle detection, velocity and feedback, and direction feedback information. 2) Task status is the sensing of the progress of the task execution sequence. This is information that indicated whether a particular stage of the task execution has been successfully completed. 3) The telerobot status is the detection of the relationship of the telerobot to its environment such as attitude information, obstacle contact, and possible other information that would be required of a particular environment. Thus the telerobot status sensing is concerned with sensing

conditions that could endanger the telerobot. 4) The device status is the feedback information required for the control of the manipulator and the manipulator/cart coordination and possible any other devices that use servos and require control (such as the sensors themselves; i.e., the camera).

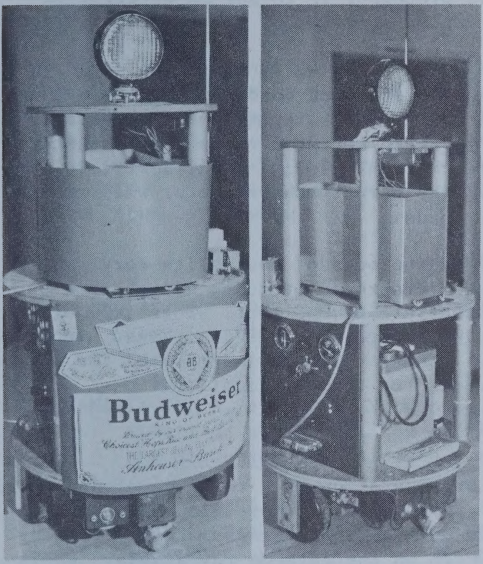


Figure 2
RALPH-with and without clothes

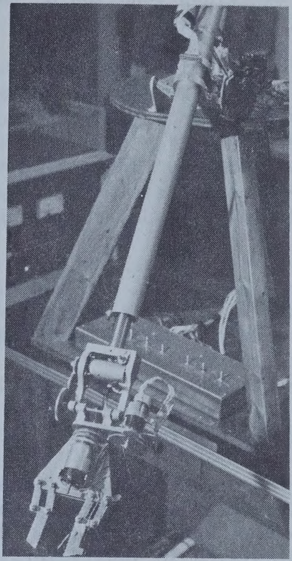


Figure 3
RALPH's Manipulator

The visual sensing is to be accomplished with a matrix diode camera and a set of appropriately placed mechanico-receptors such as potentiometers, microswitches, and pressure sensors. The matrix camera will have a digital memory to store a scene and transmit it to the mini-computer at low bandwidth transmission rates.

Onboard Processor

The onboard processor is a Motorola 6800 microprocessor computer that has approximately one thousand words of eight bit memory. There is also interfacing hardware to connect the computer to the devices and sensors. The onboard processor can be said to have four functions: a) the coding of sensor information for transmission,b) the decoding of similar data coming from the background computer, c) the automatic reflex controller, and d) error detection.

The automatic reflex system is necessary because for large distances, the response time between the operator and the telerobot could be large due to signal transmission delay times. It is also essential in the event of a telemetry failure. The onboard processor will also prevent damage to the telerobot by analyzing selected environmental data and overriding the operator's commands when priority survival situations occur. Such situations could occur when the operator did not notice a problem in time or perhaps not at all. In fact, all telerobot status parameters will be monitored by the autonomic reflex system, and appropriate behaviors will be instituted so as to maintain the status parameters within a prespecified survival range. The repertoire of autonomic reflex behaviors will include attitude control, stabilization procedures, speed monitoring, special sensor attention and others.

The final function of the onboard processor is to act in an error detection capacity. This will be implemented in terms of simple error correction codes.

The microprocessor is well suited to implement autonomic reflex responses. Microprogrammed instructions stored in PROMs (programmable read only memory) will provide excellent means to store software responses for anticipated problems. The use of stored programs in a serial fashion will to some degree handle unanticipated problems. The parent mini-computer will also be available to handle computational overloads.

Another large advantage that the microprocessor has is its very low power consumption. In addition to the microprocessor chip, all of the support electronics to the microprocessor are of MOS type. This type of semiconductor device dissipates much less power than TTL logic, and also has a better immunity to electrical noise. These characteristics make the microprocessor and MOS logic excellent devices for this application.

Operator Interface

The operator interface and monitoring station (yet to be developed) will consist of a video monitor, a cathode ray display (or oscilloscope), and a keyboard. Sensor feedback information will be processed by a mini-computer and displayed in easily intelligible form on the cathode ray display, while the visual output will be displayed unaltered on the monitor. The operator will communicate with the telerobot via the keyboard, or with a set of "joy stick" controls. The keyboard control will be realized in terms of a command language (which will be discussed in the next section). The operator will be able to access the background mini-computer for additional control resources such as stored actuator programs or detailed sensor information.

Background Computor

The background mini-computer has two basic functions. They are to process and send the operator commands, and to receive, process and display the sensory feedback data. These two primary functions will be accomplished by two programming languages -- a data display and processing language, and a command language.

Besides the operator monitoring the camera data, the information from the telerobot sensors will be sampled and organized in display form and added to the CRT display. Included in this sensor display are the status reports that will be presented to the operator concerning autonomic reflex activity, task status, telerobot status and other pertinent data. The inclusion of pattern recognition algorithms will also provide the operator with a means of automatic analysis of the sensory data.

The command language will be fairly sophisticated; however, only a limited command vocabulary will be permitted. The translation of the command into a control sequence for the actuator will be done by the background computer. The command language will also check for improper instructions.

Finally, the backgound computer could store manipulations or instructions and augment them to the computer memory. These then could be used to assist the operator at a later time.

Status of the Research

At the present time, we have the cart (as described by Hager and Northouse 1975a) functioning. The mechanical manipulator as described in Schumaker (1975) has been modified and is almost ready for interfacing to the cart. The telemetry link between the cart and the background computer has been established (Kestner and Northouse 1974) and is being updated with better equipment. The onboard processor has been acquired and is being breadboarded. The integration of most of these components is expected to be completed by November 1975.

REFERENCES

1. Chang, S. K. (1972) "On the Execution of Fuzzy Programs Using Finite-State Machines, " IEEE Trans. on Computers. Vol. C-21, No. 3, March, pp. 241-253.

2. DeMille, R. A. and R. A. Northouse, (1973) "Autonomous Computing: Perspectives and Models for Artificial Intellignece," IEEE Proc. of the Milwaukee Symposium on Automatic Control - 1974, March.

3. Eigen, D. J. and R. A. Northouse, (1972a) NCUBE: An Unsupervised Clustering Algorithm Based on Discretized Data Space, TR-AI-72-1 (November) University of Wisconsin-Milwaukee.

4. Eigen, D. J. and R. A. Northouse (1972b), NSPACE: An Unsupervised Clustering Algorithm Based on Discretized Marginal Distributions, IEEE-SMC, May 1974.

5. Ejire, Uno, Yota, Gato and Tobeyson (1972), "A Prototype Intelligent Robot that Assembles Objects from Plan Drawings," IEEE Trans. on Computers,Vol. C-21, No. 2, February, pp. 161-170.

6. Ernst, H. A. (1962), "MH-1, a Computer-Operated Mechanical Hand," Spring Joint Computer Conference, AFIPS Proc., pp. 39-51

7. Fargo, P. A. and R. A. Northouse (1974), "Character Recognition," IEEE Proc. of the Milwaukee Symposium on Automatic Control - 1974, March.

8. Forsen, G. E. (1968), "Processing Visual Data with an Automatic Eye," Pictorial Pattern Recognition, G. C. Cheng, ed., Washington D. C., Thompson, pp. 471-502.

9. Green, C. C., and B. Raphael (1968) "The Use of Theorem-Proving Techniques in Question-Answering Systems," Proc. of the 23rd ACM National Meeting, pp. 169-181.

10. Gruenberg, E. L., Ed., Handbook of Telemetry and Remote Control, McGraw-Hill Book Co: New York.

11. Guzman, A. (1968), "Decomposition of a Visual Scene into Three-Dimensional Bodies," Proc. of the Fall Joint Computer Conference, pp. 291-304.

12. Hager, J. and R. A. Northouse (1975a) "On the Design of a Robotic Cart," TR-RAIL-75-1, January), University of Wisconsin-Milwaukee.

13. --------(1975b), "A Robotic Delivery Cart," Proc. of the IEEE-MSAC-75 Conference, Milwaukee, Wisconsin (March).

14. Johnson, E. P. and W. R. Corliss (1967), Teleoperators and Human Augmentation, An AEC-NASA Technology Survey.

15. Kapp, M. and R . A. Northouse (1974), "An Unsupervised Control System for an Automatic Railroad," IEEE Proc. of the Milwaukee Symposium on Automatic Control - 1974, (March).

16. Keckler, W. G., and R. E. Larson (1970) "Control of a Robot in a Partially Unknown Environment," Automation, Vol. 6, pp. 469-476.

17. Kelly, M.D. (1970),"Edge Detection in Pictures by Computer Using Planning," Memo AIM-108, Stanford Artificial Intelligence Project, January.

18. Kestner, W. and R. A. Northouse (1974), "A Telemetry System for a Small Robotic Cart," TR-RAIL-74-21, (July), University of Wisconsin-Milwaukee.

19. Kilmer, W. L., and L.L. Sutro (1969), "Assembly of Computers to Command and Control a Robot," AFIPS Proc., Vol. 34, pp. 113-137.

20. Kocourek, C. J., and R. A. Northouse (1973), "A Robotic Cart and Some Navigation Algorithms," TR-RAIL-73-15, July, University of Wisconsin -Milwaukee.

21. Leverance, N., Kocourek, C. J., and R. A. Northouse (1973),"A Student Designed Robotic Cart," Proc. of the IEEE MSAC-73 Conference, March.

22. Leverance, N., and R. A. Northouse, "RALPH -- A Microprocessor Based Telerobot," Proc. of IEEE-SMC Conference, October 1975, San Francisco, California.

23. Manna, A., and R. Waldinger (1971), "Toward Automatic Program Synthesis," Symposium of Semantics of Algorithmic Languages, E. Engeler, ed. Springer, pp. 270-310.

24. McCarthy, Earnest, Reddy, and Vicens (1968), "A Computer with Hands, Eyes, and Ears," AFIPS Proceedings, Vol. 33.

25. Meitus, V. Y. (1969), Robots with Memory: Structure and Simulation," Progress of Cybernetics, Proc. of the First International Congress of Cybernetics, London.

26. Minsky, M.L. (1969), Semantic Information Processing, MIT Press.

27. Moore, E. T. (1959), "Shortest Path Through a Maze," Annals of the Computation Laboratory of Harvard University, Vol. 30, pp. 285-292, Harvard University Press, Cambridge, MA.

28. Nilsson, N. J. and B. Raphael (1967), "Preliminary Design of an Intelligent Robot", Computer and Information Sciences II.

29. Northouse, R. A. (1972), " An Automated Vehicular Cart," TE-EE-72-3, University of Wisconsin-Milwaukee.

30. --------, and K. S. Fu (1973), "Dynamic Scheduling of Digital Computers Using Adaptive Control and Clustering Techniques," IEEE Trans. on Systems, Man, and Cybernetics, May.

31. --------, (1973) "INAP and Inverse Nyquist," International Journal on Controls, April.

32. Pflatz, J. L., et al. (1968), "Local and Global Picture Processing by Computer," Pictorial Pattern Recognition, G. C. Cheng, ed., Washington, D. C., Thompson, pp. 353-372.

33. Pingle, K. K. (1968a), Visual Perception by a Computer," Automatic Interpretation and Classification of Images, A. Grassielli, ed., pp. 277-284.

34. --------, Singer, J. A. and W. M. Wichman (1968b), Computer Control of a Mechanical Arm Through Visual Input," Proc. of the IFIP Conference, Edinburgh.

35. Reddy, D. R. (1967), "Computer Recognition of Connected Speech," Journal of Acoustical Society of America, Vol. 42, pp. 329-347.

36. Roberts, L. G. (1965), "Machine Perception of Three Dimensional Solids," Optical and Electro-Optical Information Processing, MIT Press, Cambridge, MA, pp. 159-197.

37. Rosefeld, A. (1969), Picture Processing by Computer, Academic Press, N.Y.

38. Scheinman, V. D. and B. M. Dobrotin (1973)," A Computer Controlled Manipulator for Robot Research," Proc. of the Third International Joint Conference on Artificial Intelligence, pp. 291-297, Stanford, CA, August.

39. Schuman, P. and V. Pavelic, "A Manipulator for a Mobile Robot," Internal Report, Senior Thesis, University of Wisconsin-Milwaukee, June 1975.

40. Smith, M. H. and L.S. Coles (1973),"Design of a Low Cost, General Purpose Robot," Proc. of the Third International Joint Conference on Artificial Intelligence, pp. 330-335, August.

41. Thompson, A. (1968), "A Pathfinding Algorithm for an Autonomous Myopic Robot," Jet Propulsion Laboratory Technical Report 32-1288, Washington, D. C., December, pp. 329-338.

This research was supported in part by the National Science Foundation (grant number GK 37418) and the Graduate School of the University of Wisconsin-Milwaukee (grant numbers 7728 and 7030).

OPTIMIZATION WITH NONSTANDARD COST FUNCTIONALS

M. B. SUBRAHMANYAM and EARL D. EYMAN
Electrical Engineering, University of Iowa, Iowa City, Iowa 52242

ABSTRACT

A method of solving optimal problems involving nonstandard cost functionals is developed, where the nonstandard functional is either the quotient or the product of two definite integrals. The utilization of the theory is illustrated by means of an example.

I. INTRODUCTION

Usual formulation of optimal control problems involves cost functionals that have the localization property that if we divide a trajectory into parts and calculate the value of the functional for each part, the sum of the values of the functional for separate parts equals the value of the functional for the whole curve. Such functionals may be called standard. The majority of optimal problems involve standard functionals, but sometimes nonstandard (nonlocal) functionals do occur. For example, the abscissa of the center of mass of a curve y(x), $a \leq x \leq b$, made out of some homogeneous material is expressed by

$$J = \frac{\int_a^b x\sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx}{\int_a^b \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx} \quad . \tag{1}$$

Hence, the problem of seeking a curve for which the functional J is a minimum compared to all the other curves passing through the same points involves a nonstandard functional which is the quotient of two definite integrals.

In this paper, we will consider problems in which the cost functional is either a quotient or a product of two definite integrals. We consider the system described by

$$\dot{x} = f(x(t),u(t)) \quad , \qquad x(t_o) = x_o \tag{2}$$

where $x \in R^n$, $u \in R^m$, and the vector-valued function f is twice continuously differentiable with respect to x and is continuously differentiable with respect to u. For simplicity in proving the results, we will consider the final time t_1 as fixed and the terminal state $x(t_1)$ to be free.

An admissible control is considered to be an arbitrary piecewise continuous function which is either left continuous or right continuous at any $t \varepsilon [t_o, t_1]$ and which has values in U, a prescribed subset of R^m. Given any admissible control and an initial condition $x(t_o)$, we assume that equation (2) defines a unique continuous solution $x(t)$, $t_o \leq t \leq t_1$.

II. COST FUNCTIONAL WHICH IS THE QUOTIENT OF TWO DEFINITE INTEGRALS

Let the cost functional be of the form

$$J(u) = \frac{\int_{t_o}^{t_1} f_1(x,u)\, dt}{\int_{t_o}^{t_1} f_2(x,u)\, dt} = \frac{J_1(u)}{J_2(u)} \tag{3}$$

where $J_1(u) = \int_{t_o}^{t_1} f_1(x,u)\, dt$ and $J_2(u) = \int_{t_o}^{t_1} f_2(x,u)\, dt$. We assume that f_1 and f_2 are continuously differentiable with respect to x and u.

Let u* be an optimal control and x*(t) be the corresponding solution of (2). Let $J_2(u^*) \neq 0$. Let $x^*(t) + \delta x(t)$ be a solution of (2) with the same boundary conditions for the control $u^* + \delta u$. Then the increment

$$\Delta J = \frac{J_1(u^*) + \Delta J_1}{J_2(u^*) + \Delta J_2} - \frac{J_1(u^*)}{J_2(u^*)} = \frac{J_2(u^*)\, \Delta J_1 - J_1(u^*)\, \Delta J_2}{J_2(u^*) \left\{ J_2(u^*) + \Delta J_2 \right\}} \tag{4}$$

must be nonnegative.

Let us examine the difference ΔJ_2. We have

$$\Delta J_2 = \int_{t_o}^{t_1} [f_2(x^* + \delta x, u^* + \delta u) - f_2(x^*, u^*)]\, dt \quad . \tag{5}$$

Assuming that there exists a uniform bound for responses of (2) on $[t_o, t_1]$ for all variations δu whose values lie in any arbitrary compact set contained in U, we have

$$|\Delta J_2| \leq \int_{t_o}^{t_1} K \left(\sum_{i=1}^{n} |\delta x^i| \right) dt + \int_{t_o}^{t_1} K \left(\sum_{I+1}^{m} |\delta u^i| \right) dt \tag{6}$$

where K is the Lipschitz constant. Also, we have

$$\delta x^i(t) = \int_{t_o}^{t} [f^i(x^* + \delta x, u^* + \delta u) - f^i(x^*, u^*)]\, ds \quad . \tag{7}$$

Let M be the maximum of Lipschitz constants for the f^i. Then it can be easily shown that

$$\sum_{i=1}^{n} |\delta x^i(t)| \leq Mn\ e^{Mn(t_1-t_o)} \int_{t_o}^{t_1} \left(\sum_{i=1}^{m} |\delta u^i| \right) dt \quad . \tag{8}$$

From (6) and (8), it follows that

$$|\Delta J_2| \leq K \left\{ Mn(t_1-t_o)\ e^{Mn(t_1-t_o)} + 1 \right\} \int_{t_o}^{t_1} \left(\sum_{i=1}^{m} |\delta u^i| \right) dt \tag{9}$$

which tends to zero as $\delta u \to 0$.

For variations which have sufficiently small values for $\int_{t_o}^{t_1} \left(\sum_{i=1}^{m} |\delta u^i| \right) dt$, ΔJ_2 can be made so small that $J_2(u^*) + \Delta J_2$ has the same sign as $J_2(u^*)$. Hence the denominator $J_2(u^*)\ \{J_2(u^*) + \Delta J_2\}$ of equation (4) is positive and we have

$$J_2(u^*)\ \Delta J_1 - J_1(u^*)\ \Delta J_2 \geq 0 \quad . \tag{10}$$

Letting $\dfrac{J_1(u^*)}{J_2(u^*)} = \lambda$, we have

$$\Delta(J_1 - \lambda J_2) \geq 0 \quad . \tag{11}$$

III. THE MAXIMUM CONDITION

We will now show that $u^*(t)$ should satisfy the maximum principle for the intermediate functional $J_1 - \lambda J_2$. Let $\Omega(t_o,t_1)$ denote the space of all admissible controls on $[t_o,t_1]$ with norm defined by $||u|| = \int_{t_o}^{t_1} \left(\sum_{i=1}^{m} |u^i(t)| \right) dt$. We will define the Hamiltonian H as

$$H(\psi,x,u) = \sum_{i=o}^{n} \psi_i\ f^i(x,u) \tag{12}$$

where $f^o = f_1 - \lambda f_2$, $\dot{\psi}_o = 0$, $\psi_o(t_1) = -1$, and $\dot{\psi}_i = -\dfrac{\partial H}{\partial x^i}$, $\psi_i(t_1) = 0$, $i = 1,2,\ldots,n$. Following similar reasoning as in the appendix of [1], it can be derived that

$$\Delta(J_1 - \lambda J_2) = -\int_{t_o}^{t_1} [H(\psi,x^*,u^* + \delta u) - H(\psi,x^*,u^*)]\, dt - O(\|\delta u\|) \quad (13)$$

where $\psi(t)$ corresponds to $u^*(t)$ and $\frac{O(\|\delta u\|)}{\|\delta u\|} \to 0$ as $\|\delta u\| \to 0$. We have to prove that $H(\psi(t),x^*(t),u^*(t)) \geq H(\psi(t),x^*(t),u)$ for all $u\varepsilon U$ and for all $t\varepsilon[t_o,t_1]$.

Suppose that the above maximum condition does not hold at some $\bar{t}\varepsilon[t_o,t_1]$. Then there exists a $\bar{u}\varepsilon U$ such that

$$H(\psi(\bar{t}),x^*(\bar{t}),\bar{u}) > H(\psi(\bar{t}),x^*(\bar{t}),u^*(\bar{t})) \quad . \quad (14)$$

In view of the piecewise continuity of u* and the continuity of x(t), $\psi(t)$, and $H(\psi,x,u)$, it follows that there is an interval [t',t"] containing $\bar{t}$ and $\varepsilon > 0$ such that

$$H(\psi(t),x^*(t),\bar{u}) - H(\psi(t),x^*(t),u^*(t)) > \varepsilon \qquad \text{for all} \qquad t\varepsilon[t',t"] .$$

Let u(t) be the piecewise continuous function which is equal to u*(t) outside [t',t"] and $\bar{u}$ on [t',t"]. It is obvious that $\|u - u^*\| \to 0$ as $(t" - t') \to 0$. Using (13), we have

$$\Delta(J_1 - \lambda J_2) < -\varepsilon(t" - t') + |O(\|u - u^*\|)| \quad . \quad (15)$$

By selecting [t',t"] sufficiently small, $\Delta(J_1 - \Delta J_2)$ can be made negative, which contradicts (11). Hence, we have proved that u*(t) satisfies the maximum principle for the intermediate functional $J_1 - \lambda J_2$.

IV. COST FUNCTIONAL WHICH IS THE PRODUCT OF TWO DEFINITE INTEGRALS

Let us consider the cost functional of the form

$$J(u) = \left(\int_{t_o}^{t_1} f_1(x,u)\, dt\right) \left(\int_{t_o}^{t_1} f_2(x,u)\, dt\right) = J_1(u)\ J_2(u) \quad (16)$$

where the functions f_1 and f_2 possess continuous partial derivatives with respect to x and u, and $J_1(u) = \int_{t_o}^{t_1} f_1(x,u)\, dt$.

Consider an optimal control u*(t) and its corresponding trajectory x*(t). Let x*(t) + δx(t) be the trajectory corresponding to the varied control u* + δu. We have

$$0 \leq \Delta J = (J_1 + \Delta J_1)(J_2 + \Delta J_2) - J_1 J_2 = J_1\, \Delta J_2 + J_2\, \Delta J_1 + \Delta J_1\, \Delta J_2 . \tag{17}$$

Neglecting $\Delta J_1\, \Delta J_2$ compared to the other terms for small variations of u* ($\Delta J_1\, \Delta J_2$ can be neglected since $\Delta J_1\, \Delta J_2 = 0(\|\delta u\|)$, and assuming $J_2(u) \neq 0$, we have

$$\Delta(J_1 + \lambda J_2) \geq 0 \tag{18}$$

where $\lambda = J_1(u^*)/J_2(u^*)$. As in Section III, it can be proved that u* should satisfy the maximum principle for the intermediate functional $J_1 + \lambda J_2$.

V. AN EXAMPLE

To illustrate the utilization of the theory, we will solve the following problem: find a curve $x^1(t)$ which yields the minimum of the expression

$$J = \frac{\int_0^1 (\dot{x}^1)^2\, dt}{\int_0^1 (x^1)^2\, dt} \tag{19}$$

under the initial condition $x^1(0) = 0$.

Letting

$$\dot{x}^1 = u(t) \quad , \tag{20}$$

we have $J = \int_0^1 u^2\, dt \Big/ \int_0^1 (x^1)^2\, dt$. The intermediate functional is $\int_0^1 \{u^2 - \lambda (x^1)^2\}\, dt$. The Hamiltonian is given by

$$H = \psi_1 u - u^2 + \lambda (x^1)^2$$

where $d\psi_1/dt = -\, 2\, \lambda x^1$ with the boundary condition $\psi_1(1) = 0$. Applying the maximum principle, we get the following optimal control:

$$u(t) = \frac{\psi_1(t)}{2} = - \int_0^t \lambda x^1(s)\, ds \quad . \tag{21}$$

Hence, from Eqs. (20) and (21), we see that the curve satisfies $\ddot{x}^1 + \lambda x^1 = 0$, whose solution is $x^1(t) = A \sin(\sqrt{\lambda}\, t) + B \cos(\sqrt{\lambda}\, t)$ where A and B are constants which can be determined from the boundary conditions.

The curves which satisfy the boundary conditions are

$$x^1(t) = A \sin(\sqrt{\lambda}\, t) \quad ,$$

where $\lambda = n^2\pi^2/4$, $n = 1,3,5,7,\ldots$, and $A = \text{constant} \neq 0$. Let us compute the value of J for these curves:

$$J = \frac{\int_0^1 A^2 \frac{n^2\pi^2}{4} \cos^2\left(\frac{n\pi}{2} t\right) dt}{\int_0^1 A^2 \sin^2\left(\frac{n\pi}{2} t\right) dt} = \frac{n^2\pi^2}{4} \quad . \tag{22}$$

The least possible value of J is achieved for $n = 1$ when $x^1(t) = A \sin\left(\frac{\pi t}{2}\right)$.

VI. CONCLUSIONS

A method of solving problems involving nonstandard cost functionals is developed, where the nonstandard functional is either the quotient or the product of two definite integrals. An example is given to illustrate the utilization of the theory.

REFERENCES

[1] Rozonoér, L. I., "L. S. Pontryagin's Maximum Principle in the Theory of Optimum Systems, I," Automation and Remote Control, 20(10), 1959, pp. 1288-1302.

[2] Petrov, Iu. P., Variational Methods in Optimum Control Theory, Academic Press, New York, 1968.

[3] Luenberger, D. G., Optimization by Vector Space Methods, John Wiley & Sons, Inc., New York, 1969.

APPLICATION OF MATHEMATICAL PROGRAMMING TO OPTIMAL CONTROL PROBLEMS WITH UNORTHODOX COST FUNCTIONALS AND CONSTRAINTS: A SUMMARY OF RESULTS

DIVAKARUNI P. SUDHAKAR† and EARL D. EYMAN
Electrical Engineering, The University of Iowa, Iowa City, Iowa 52242

ABSTRACT

Several unorthodox control problems are considered with relaxed assumptions on the differentiability requirements on the cost functional and other side constraints. It is found that the gradients of the cost functional and other constraints can be replaced by their Gateaux differentials in the adjoint equation. Necessary conditions are derived for optimality in the form of an adjoint equation and a pointwise maximal condition. The techniques of Mathematical Programming and the Dubovitskii-Milyutin Theorem are employed.

I. INTRODUCTION

Three control problems are considered in this article: (1) those with non-differentiable cost functionals which can arise in the application of control theory to problems in economics, (2) those with non-differentiable side constraints which arise in bounded-phase control when the phase restriction has corners, and (3) those with non-differentiable isoperimetric constraints which provide an effective approximation to bounded phase control with corners [5].

When a function is continuous and convex, it is Gateaux differentiable everywhere in every direction [8]. Hence the assumptions of continuity and convexity are made on the functionals and the constraints to ensure the existence of the Gateaux differentials. The assumptions made on the form of the Gateaux differential are the same as that of Luenberger [6] and this paper presents an extension of the results [6] to control problems of a more general nature. The results are extended to include problems involving bounded controls, free terminal time and terminal state conditions in the form of smooth manifolds. The theorem due to Dubovitskii and Milyutin [3] and their ingenious transformation [2] are employed.

The concepts of the set (cone) of feasible directions, the set (cone) of directions of decrease and the set of tangent directions [4] are relied on heavily to permit the application of the Dubovitskii-Milyutin Theorem. Since the purpose of this paper is to summarize the results obtained by the authors, no elaborate details of the proofs are given. The mathematical details are limited to the description of the different cones mentioned earlier and their duals. The detailed proofs are furnished in [1].

---

†Presently with the Division of Mathematical Sciences, The University of Iowa, Iowa City, Iowa 52242.

II. NOTATION

The symbol $C^k([t_o,T],R^n)$ denotes the space of n-dimensional functions defined on $[t_o,T]$ that are k-times continuously differentiable. The familiar terminology like "of class C^1" is also used. The symbol $L^p([t_o,T],R^m)$ denotes the m-dimensional L^p (Lebesgue) functions. The symbol $\langle \cdot,\cdot \rangle$ denotes the inner product of two elements. Notationally no distinction is made between an element and its transpose. The symbol $\delta^+ g(x;h)$ denotes the Gateaux differential of the function g at x in the direction h.

III. CONTROL PROBLEMS WITH NON-DIFFERENTIABLE COST FUNCTIONALS

We consider a control process in R^n

$$\dot{x} = f(x,u) \tag{3.1}$$

$$x(t_o) = a \tag{3.2}$$

$$x(t_1) = b \tag{3.3}$$

where f is of class C^1 with respect to its arguments. The control function u belongs to the class of all m-dimensional essentially bounded (L^∞) functions. It is assumed that

$$u(t) \varepsilon M \subseteq R^m \tag{3.4}$$

where M is closed and convex. The task is to find a control u in the indicated admissible class of functions such that the cost functional

$$\phi(x(\cdot),u(\cdot)) = \int_{t_o}^{T} \ell(x(t),u(t))dt \tag{3.5}$$

for a given T is a minimum. The function ℓ is assumed to be continuous, convex but not necessarily differentiable. However, it is Gateaux differentiable and the Gateaux differential of ℓ at x in the direction h denoted by $\delta^+(x;h)$ is assumed to be of the form [6]

$$\delta^{+}\ell(x;h) = \ell_x^{*}\, h + \sum_{i=1}^{n} |y_{x_i}^{*}\, h| \tag{3.6}$$

where ℓ_x^{*} and $y_{x_i}^{*}$ are n-vectors. It is assumed that the function ℓ is differentiable with respect to u.

Theorem 3.1: Let (x_o,u_o) be an optimal response-control pair for the problem (3.1) through (3.6) (note that T is fixed). Then there exist (i) an n-vector function s(t) of bounded variation on $[t_o,T]$ with the property $|s(t)| = \max_{1\le i\le n} |s_i(t)| \le 1$, (ii) a constant $c\varepsilon R^n$, (iii) a real number $\lambda_o \ge 0$, and (iv) an n-vector $\psi(t)$ such that the following are satisfied:

1. $\dot{\psi}(t) = f_x(x_o(t),u_o(t))\psi(t) + \lambda_o[\ell_x^{*}(t) + s(t)Y]\ , \qquad \psi(T) = c\ ,$ (3.7)

2. $f_u(x_o(t),u_o(t))\psi(t) + \lambda_o\ \ell_u(x_o(t),u_o(t)),\ u - u_o(t)\ \ge 0$ (3.8)

   for all $u\varepsilon M$ almost everywhere on $[t_o,T]$ and

3. λ_o and $\psi(t)$ are not zero simultaneously.

In equation (3.7)

$$Y = \begin{pmatrix} y_{x_1}^{*} \\ \vdots \\ y_{x_n}^{*} \end{pmatrix}$$

is an n × n matrix.

Proof: By virtue of the assumption (3.6), we have

$$\delta^{+}\phi(x_o(\cdot),u_o(\cdot);h(\cdot),v(\cdot)) = \int_{t_o}^{T} \delta^{+}\ell(x_o(t),u_o(t);h(t),v(t))dt$$

$$= \int_{t_o}^{T} \delta^{+}\ell(x_o(t),u_o(t);h(t),0)dt$$

$$+ \int_{t_o}^{T} \delta^{+}\ell(x_o(t),u_o(t);0,v(t))dt$$

$$= \int_{t_o}^{T} \ell_x^{*}(t)h(t) + \sum_{i=1}^{n} \left|y_{x_i}^{*}(t)h(t)\right|dt$$

$$+ \int_{t_o}^{T} \ell_u(x_o(t),u_o(t))v(t)dt$$

$$= \int_{t_o}^{T} \ell_x^{*}(t)h(t) + \sum_{i=1}^{n} \max_{\substack{|s_i(t)|\le 1\\ 1\le i\le n}} s_i(t)y_{x_i}^{*}(t)h(t)dt$$

$$+ \int_{t_o}^{T} \ell_u(x_o(t),u_o(t))v(t)dt \quad . \tag{3.9}$$

Since the maximization in (3.9) is pointwise, one has

$$\delta^{+}\phi(x_o(\cdot),u_o(\cdot);h(\cdot),v(\cdot) = \max_{\substack{|s_i(t)|<1\\ 1\le i\le n}} \int_{t_o}^{T} \ell_x^{*}(t)h(t) + \sum_{i=1}^{n} s_i(t)y_{x_i}^{*}(t)h(t)$$

$$+ \int_{t_o}^{T} \ell_u(x_o(t),u_o(t))v(t)dt \quad .$$

In matrix notation, the above equation can be written as

$$\delta^{+}\phi(x_o(\cdot),u_o(\cdot);h(\cdot),v(\cdot)) = \max_{|s(t)|<1} \int_{t_o}^{T} \ell_x^{*}(t)h(t) + s(t)Y\,h(t)dt$$

$$+ \int_{t_o}^{T} \ell_u(x_o(t),u_o(t))v(t)dt \quad . \tag{3.10}$$

By Theorem 2.2 in Chapter II of [8]

$$\delta^{+}\phi(x_o(\cdot),u_o(\cdot);h(\cdot),0) = \max_{x^{*}\in M(x_o(\cdot))} x^{*}(h) \quad . \tag{3.11}$$

By the Riesz Representation Theorem [7] x^* is a function of bounded variation. From (3.10) and (3.11) it is clear that s(t) is a function of bounded variation. The rest of the proof is dependent on the calculation of different convex cones (cone of directions of decrease, etc.) and their duals. The dual cones are essentially the support functionals under convexity assumptions. The conditions (3.7) and (3.8) follow from the application of the Dubovitskii-Milyutin Theorem and Lusin's Theorem [7]. The details are furnished in [1].

The next theorem is a generalization of Theorem 3.1 to problems with free terminal time and arbitrary control restraints. The assumptions on f and ℓ are slackened further.

Theorem 3.2: Consider the problem (3.1) through (3.6) except that (i) T is free, (ii) M is any arbitrary bounded set in R^n and (iii) ℓ and f are continuous in u but not necessarily differentiable. Let (x_o, u_o, T^*) be an optimal solution to the new problem. Then there exist λ_o, s(t) and $\psi(t)$ defined on $[t_o, T^*]$ as in Theorem 3.1 with the following modifications:

1. The terminal condition in (3.7) becomes $\psi(T^*) = c$.
2. Replace the maximal condition (3.8) by

$$H(x_o(t),u_o(t),\psi(t)) = \max_{u\varepsilon M} H(x_o(t),u,\psi(t)) = 0 \tag{3.12}$$

almost everywhere on $[t_o, T^*]$ for all $u\varepsilon M$ where

$$H(x_o(t),u,\psi(t)) \triangleq \langle f(x_o(t),u),\psi(t)\rangle + \lambda_o \ell(x_o(t),u) \quad . \tag{3.13}$$

IV. CONTROL PROBLEMS WITH NON-DIFFERENTIABLE SIDE CONSTRAINTS

Consider a control process described by (3.1) through (3.5) except that ℓ is of class C^1 with respect to its arguments together with the side constraint

$$g(x(t) \leq 0 \tag{4.1}$$

for all $t_o \leq t \leq T$, where T is fixed and g is continuous, convex but not necessarily differentiable in its argument. However it is Gateaux differentiable. Assume further that $\delta^+ g(x(t);h(t))$ is of the form

$$\delta^+ g(x(t);h(t)) = g_x^*(t)h(t) + \sum_{i=1}^{n} |G_{x_i}^*(t)h(t)| \tag{4.2}$$

where g_x^* and $G_{x_i}^*$ are n-vectors.

Theorem 4.1: Let (x_o,u_o) be a solution to the problem described above. Then there exist (i) a real number $\lambda_o \geq 0$, (ii) a vector $\psi(t)$, (iii) a non-negative measure $\mu(t)$ concentrated on the set $\Gamma = \{t\varepsilon[t_o,T] | g(x_o(t) = 0\}$ (iv) a constant $c\varepsilon R^n$ and (v) an n-vector function $\nu(t)$ with the property $|\nu(t)| = \max_{1\leq i\leq n} |\nu_i(t)| \leq 1$ such that the following are satisfied:

1. $$\psi(t) = -c + \int_t^T - f_x(x_o(s),u_o(s))\psi(s) + \lambda_o \ell_x(x_o(s),u_o(s))ds$$

$$+ \int_t^T [g_x^*(s) + \nu(s)G_x^*(s)]d\mu(s) \tag{4.3}$$

where $G_x^*(s) = \begin{bmatrix} G_{x_1}^* \\ \vdots \\ G_{x_n}^*(s) \end{bmatrix}$ is an n × n matrix,

2. $$\left\langle f_u(x_o(s),u_o(s))\psi(s) + \lambda_o \ell_u(x_o(s),u_o(s)), u - u_o(s)\right\rangle \geq 0 \tag{4.4}$$

almost everywhere on $[t_o,T]$ for all $u\varepsilon M$, and

3. $\lambda_o, \psi(t)$ and $d\mu(t)$ are not simultaneously zero.

Proof: Define the functional

$$G(x(\cdot)) = \max_{t_o\leq t\leq T} g(x(t)) \quad .$$

Let $Q = \{x(\cdot)\varepsilon C([t_o,T],R^n) | G(x(\cdot)) \leq 0\}$.

The set of feasible directions for Q at $x_o(\cdot)$ is given by

$$K = \{h(\cdot)\varepsilon C([t_o,T],R^n) | \delta^+ G(x_o(\cdot);h(\cdot)) < 0\}$$

where

$$\delta^+ G(x_o(\cdot);h(\cdot)) = \max_{t\varepsilon\Gamma} \delta^+ g(x_o(t);h(t))$$

$$= \max_{t_o\leq t\leq T} \delta^+ g(x_o(t);h(t))$$

and Γ is as defined earlier. Any functional $f_1 \varepsilon K^*$, the dual cone of K has a representation

$$f_1(h) = \int_{t_o}^{T} h(t)d\mu(t) = \int_{\Gamma} h(t)d\mu(t)$$

where $\mu(t)$ is a non-decreasing function of bounded variation and hence a non-negative measure. It is supported on the set $\Gamma = \{t\varepsilon[t_o,T] | g(x(t)) = 0\}$. The rest of the proof is similar to the proof of Theorem 3.1.

The next theorem is a generalization of Theorem 4.1 to the case of free-terminal time.

Theorem 4.2: Consider the control problem of Theorem 4.1 except that T is free and ℓ and f are continuous in u, but not necessarily differentiable. If (x_o,u_o,T^*) is an optimal solution to the new problem, then the following modifications are to be made in the statement of Theorem 4.1:

1. Condition 1 of Theorem 4.1 holds with T replaced by T^*. The same is true of the domain of definition of the functions μ, s and ψ.

2. The condition 2 of Theorem 4.1 is replaced by

$$H(x_o(t),u_o(t),\psi(t)) = \max_{u\varepsilon M} H(x_o(t),u,\psi(t)) = 0 \tag{4.5}$$

almost everywhere on $t_o \leq t \leq T^*$ where

$$H(x_o(t),u,\psi(t)) \triangleq \left\langle f(x_o(t),u),\psi(t)\right\rangle + \lambda_o \ell(x_o(t),u) \quad . \tag{4.6}$$

V. CONTROL PROBLEMS WITH UNORTHODOX ISOPERIMETRIC CONSTRAINTS

We consider the differential process (3.1) through (3.4). The objective is to find a control $u\varepsilon L^\infty([t_o,T],R^m)$ such that the cost functional

$$\phi(x(\cdot),u(\cdot)) = \int_{t_o}^{T} \ell_1(x(t),u(t))dt \tag{5.1}$$

is minimized for a given T while the condition

$$g(x(\cdot),u(\cdot)) = \int_{t_o}^{T} \ell_2(x(t),u(t))dt \leq 0 \tag{5.2}$$

is satisfied. The constraint (5.2) is referred to as the "isoperimetric constraint." The function ℓ_1 is assumed to be of class C^1 with respect to its arguments. The function ℓ_2 is assumed to be continuous, convex but not necessarily differentiable in x. However, it is Gateaux differentiable and the Gateaux differential of ℓ_2 is assumed to be of the form

$$\delta^+\ell_2(x,u;h,0) = \ell^*_{2x}h + \sum_{i=1}^{n} |y^*_{2x_i}h| \tag{5.3}$$

where ℓ^*_{2x} and $y^*_{2x_i}$ are n-vectors. The function ℓ_2 is assumed to be differentiable in u.

Theorem 5.1: Let (x_o,u_o) be an optimal solution to the control process described by (3.1) through (3.4) and (5.1) through (5.3). Then there exist (i) real numbers $\lambda_1,\lambda_2 \geq 0$, (ii) a constant $c\varepsilon R^n$, (iii) an n-vector $\psi(t)$ and (iv) an n-vector function $s_o(t)$ of bounded variation on $[t_o,T]$ such that the following are satisfied:

1. $$\dot{\psi}(t) = -f_x(x_o(t),u_o(t))\psi(t) + \lambda_1\ell_{1x}(x_o(t),u_o(t))$$
$$+\lambda_2[\ell^*_{2x}(t) + s_o(t)Y_2] \quad , \qquad \psi(T) = c \tag{5.4}$$

2. $$\left\langle f_u(x_o(t),u_o(t)\psi(t) + \sum_{i=1}^{2} \lambda_i\ell_{iu}(x_o(t),u_o(t)), u - u_o(t)\right\rangle \geq 0 \tag{5.5}$$

   for all $u\varepsilon M$ almost everywhere on $[t_o,T]$, and

3. λ_1,λ_2 and $\psi(t)$ are not simultaneously zero.

In equation (5.4) Y_2 is an n × n matrix composed of the vectors $y^*_{2x_i}, i=1,\ldots,n$.

Theorem 5.2: Consider the control process considered in Theorem 5.1 except that (i) T is free and, (ii) the functions f, ℓ_1 and ℓ_2 are continuous in u, but not necessarily differentiable. If (x_o,u_o,T^*) is a solution to the new problem, then the following modifications are made to the necessary conditions stated in Theorem 5.1:

1. The function $s_o(t)$ of bounded variation is defined on $[t_o,T^*]$.

2. The differential equation has a terminal condition $\psi(T^*) = c$.

3. Replace the maximal condition (5.5) by the following

$$H(x_o(t),u_o(t)\psi(t)) = \max_{u\varepsilon M} H(x_o(t),u,\psi(t)) = 0 \tag{5.6}$$

almost everywhere on $[t_o,T^*]$ where

$$H(x_o(t),u,\psi(t)) = \left\langle f(x_o(t),u),\psi(t)\right\rangle + \sum_{i=1}^{2} \lambda_i \ell_i(x_o(t),u) \quad . \tag{5.7}$$

VI. TERMINAL STATE CONDITIONS IN THE FORM OF SMOOTH MANIFOLDS

In Sections III, IV and V the terminal state constraints on x can be of the form

$$x(T) \varepsilon H(x(T))$$

where H is a smooth k-manifold in R^n. Moreover, the matrix $H'(x_o(T))$ has a rank k. Then the necessary conditions are essentially the same as those presented above except for the terminal condition on the adjoint $\psi(t)$ which is of the form

$$\psi(T) = \sum_{i=1}^{k} \eta_i H_i'(x_o(T))$$

where η_i, i = 1,...,k are constants. If the initial conditions also involve smooth manifolds, they can be handled in a similar manner.

VII. CONCLUSIONS

Necessary conditions for several optimal control problems with relaxed assumptions on the differentiability assumptions are presented. It is beyond the scope of this paper to furnish all the mathematical details which can be found in [1]. The Gateaux differential operators of the non-differentiable functions replace the gradients in the adjoint equations.

REFERENCES

[1] Divakaruni, S. P., "Mathematical Programming and Unorthodox Control Problems," Ph.D. Dissertation, The University of Iowa, Iowa City, Iowa, 1975.

[2] Dubovitskii, A. Ya. and Milyutin, A. A., "Extremum Problems in the Presence of Constraints," Zh. Vychisl. Mat. i Mat. Fiz., Vol. 5, 1965.

[3] Dubovitskii, A. Ya. and Milyutin, A. A., "Extremum Problems in the Presence of Constraints," DOKO. Akad. Nauk SSSR, Vol. 149, 1963.

[4] Girsnov, I. V., "Lectures on Mathematical Theory of Extremum Problems," Lecture Notes in Economics and Mathematical Systems, Vol. 67, Springer-Verlag, New York, 1972.

[5] Lee, E. B., "An Approximation to Linear Bounded Phase Coordinate Control Problems," J. of Math. Anal. and Appl., Vol. 13, 1966.

[6] Luenberger, D. G., "Control Problems with Kinks," IEEE Trans. Automatic Control, Vol. AC-15, 1970.

[7] Natanson, I. P., Theory of Functions of a Real Variable, Frederick Ungar, New York, 1962

[8] Pschenichinyi, B. N., Necessary Conditions for an Extremum, Marcel Dekker, Inc., New York, 1971.

NEW RESULTS IN THE LEAST PTH APPROACH TO MINIMAX DESIGN

J.W. BANDLER
Group on Simulation, Optimization and Control
McMaster University, Hamilton, Canada

C. CHARALAMBOUS
Department of Combinatorics and Optimization
University of Waterloo, Waterloo, Canada

J.H.K. CHEN and W.Y. CHU
Bell-Northern Research
Ottawa, Canada

ABSTRACT

We present results of two general approaches for obtaining minimax designs through a sequence of least pth approximations demonstrating increased efficiency over previous least pth algorithms. Documented computer programs are available.

INTRODUCTION

This paper demonstrates the acceleration of convergence to minimax solutions by extrapolation on a sequence of least pth solutions [1] with geometrically increasing values of p, and compares the results with an efficient extension of work by Charalambous and Bandler [2,3], in which a sequence of least pth solutions with finite values of p are obtained in an effort to reach a minimax solution. Documented computer programs are available [4,5], as well as the theoretical background [6,7,8].

THEORY

We minimize w.r.t. $\boldsymbol{\phi}$ for given ξ and $p > 1$ the function

$$U(\boldsymbol{\phi},\xi,p) \triangleq \begin{cases} M(\boldsymbol{\phi},\xi)\left(\sum\limits_{i\in K}\left(\dfrac{f_i(\boldsymbol{\phi}) - \xi}{M(\boldsymbol{\phi},\xi)}\right)^q\right)^{\frac{1}{q}} & \text{for } M(\boldsymbol{\phi},\xi) \neq 0 \\ 0 \text{ for } M(\boldsymbol{\phi},\xi) = 0 & \end{cases} \quad (1)$$

where $M_f(\boldsymbol{\phi}) \triangleq \max\limits_{i\in I} f_i(\boldsymbol{\phi})$, $M(\boldsymbol{\phi},\xi) \triangleq M_f(\boldsymbol{\phi}) - \xi$, $q \triangleq p \operatorname{sgn} M(\boldsymbol{\phi},\xi)$ and

$$K = \begin{cases} I \subset \{1,2,\ldots,m\} & \text{for } M(\boldsymbol{\phi},\xi) < 0 \\ J \triangleq \{i \mid f_i(\boldsymbol{\phi}) - \xi \geq 0,\ i \in I\} & \text{for } M(\boldsymbol{\phi},\xi) > 0 \end{cases} \quad (2)$$

and where $\boldsymbol{\phi} \triangleq [\phi_1 \ \phi_2 \ . \ . \ . \ \phi_k]^T$ is the design parameter vector, and $f_1(\boldsymbol{\phi})$, $f_2(\boldsymbol{\phi})$, ..., $f_m(\boldsymbol{\phi})$ are m linear or nonlinear functions directly related to the response error functions such that if $M_f(\boldsymbol{\phi}) > 0$ the specifications are violated and if $M_f(\boldsymbol{\phi}) < 0$ the specifications are satisfied.

Charalambous has shown [6] that if we have $\boldsymbol{u}$ and $\boldsymbol{\phi}$ such that

---

This work was supported by the National Research Council of Canada under Grant A7239.

$$\begin{aligned} &\sum_{i=1}^{m} u_i \; \underset{\sim}{\nabla} f_i(\underset{\sim}{\phi}) = \underset{\sim}{0} \\ &\sum_{1=1}^{m} u_i = 1 \end{aligned} \qquad (3)$$

$$u_i \geq 0, \; i = 1,2,\ldots,m$$

then, if $\sum_{i=1}^{m} u_i f_i(\underset{\sim}{\phi})$ is convex w.r.t. $\underset{\sim}{\phi}$,

$$\sum_{i=1}^{m} u_i f_i(\underset{\sim}{\phi}) \leq M_f(\check{\underset{\sim}{\phi}}) \leq M_f(\underset{\sim}{\phi}) \qquad (4)$$

where $\check{\underset{\sim}{\phi}}$ is the minimax optimum which is being sought, $\underset{\sim}{u} \overset{\Delta}{=} [u_1 \; u_2 \; \ldots \; u_m]^T$ and $\underset{\sim}{\nabla} \overset{\Delta}{=} [\partial/\partial\phi_1 \;\; \partial/\partial\phi_2 \; \ldots \; \partial/\partial\phi_k]^T$. The conditions (3) are satisfied at each optimum point $\check{\underset{\sim}{\phi}}(p,\xi)$ for a least pth objective function, yielding

$$\sum_{i=1}^{m} u_i f_i(\check{\underset{\sim}{\phi}}(p,\xi)) \leq M_f(\check{\underset{\sim}{\phi}}) \leq M_f(\check{\underset{\sim}{\phi}}(p,\xi)) \qquad (5)$$

where, assuming K contains all critical sample points,

$$u_i = \frac{v_i}{\sum_{i \varepsilon K} v_i} \qquad (6)$$

$$v_i = \begin{cases} \left(\dfrac{f_i(\check{\underset{\sim}{\phi}}(p,\xi)) - \xi}{M(\check{\underset{\sim}{\phi}},\xi)} \right)^{q-1} & \text{for } i \; \varepsilon \; K \\ 0 \text{ for } i \notin K \end{cases} \qquad (7)$$

The first term of (5) is, under the stated conditions, a lower bound on $M_f(\check{\underset{\sim}{\phi}})$. It is, at any least pth solution, an optimistic indication of the ultimate minimax error to be expected for a particular design.

The ξ-Algorithm (an implementation of work by Charalambous [6])

ξ^{r+1} is set to the lower bound from the rth optimization. p is kept constant. We start at $\underset{\sim}{\phi}^0$ by setting ξ^1 to an initial guess of a lower bound and $\xi^0 < \xi^1$ to a margin for discarding all functions f_i for which $f_i(\underset{\sim}{\phi}^0) < \xi^0$, which are considered unlikely to be active at $\check{\underset{\sim}{\phi}}$. ξ^r is used as a level for discarding functions, so that $I^{r+1} = \{i \,|\, f_i(\check{\underset{\sim}{\phi}}^r) \geq \xi^r\}$. The functions specifically required for the least pth objective and its gradients are successively reduced as the optimum is approached enabling a saving of effort in gradient computations. The program, called MINOPT [4], can also be restarted efficiently from any point instead of having to repeat the entire process. Convergence of the basic algorithm has been proved [6].

The p-Algorithm (an implementation of work by Chu [8])

p_{r+1} is set to cp_r, where $c > 1$. ξ is kept constant. We let $\phi_0^r = \check{\phi}^r$ and for $r > 1$,

$$\phi_j^r = \frac{c^j \phi_{j-1}^r - \phi_{j-1}^{r-1}}{c^j - 1}, \quad j=1,\ldots,n_0, \tag{8}$$

where $n_0 = \min\,[r-1, n]$, ϕ_j^r signifies the jth order estimate of $\check{\phi}$ after r minima have been obtained, and n is the highest order of extrapolation.

The next minimum is estimated by solving for $n_0 \geq 1$

$$\phi_{j-1}^{r+1} = \frac{(c^j - 1)\phi_j^{r+1} + \phi_{j-1}^r}{c^j} \tag{9}$$

for $j = n_0, \ldots, 1$, using $\phi_{n_0}^{r+1} = \phi_{n_0}^r$. The next starting point is ϕ_0^{r+1}.

We take $I^{r+1} = \{i \mid \mu_i > \eta\}$ where η is preassigned and

$$\mu_i = \frac{v_i}{\sum\limits_{i \in K} v_i} \tag{10}$$

$$v_i = \begin{cases} \left(\dfrac{f_i(\phi_0^{r+1}) - \xi}{M(\phi_0^{r+1}, \xi)} \right)^{q_{r+1}} & \text{for } i \in K \\ 0 \text{ for } i \notin K & \end{cases} \tag{11}$$

and K is chosen according to (2) with I set to I^r.

p is not increased when r = 1 in determining μ_i. No extrapolation can be made at this stage so the starting point for the next minimization is the solution to the first one. In the program, called FLOPT2 [5], past solutions may be retained for future runs permitting extrapolation to be implemented immediately. Theoretical justification of the algorithm has been demonstrated [7,8].

EXAMPLES

The unconstrained minimization method throughout was a recent quasi-Newton method [9], and the computer used was a CDC 6400.

We consider the same two starting points (leading to Problems 1 and 2) and sample points for the optimization in the minimax sense of a three-section 100 percent relative bandwidth 10:1 transmission-line transformer as in Table I of Charalambous and Bandler [2]. The effort to reach or exceed a reflection coefficient of 0.19729 (optimal to 5 figures) is compared taking f_i as the modulus of the reflection coefficient. We take as variables the lengths ℓ_i and characteristic impedances Z_i. About 1/3 to 1/2 of the response evaluations used by the Charalambous-Bandler algorithms [2], tabulated in Table I, are required. Increasing p in the ξ-algorithm gave poorer results. The p-algorithm appears relatively insensitive to the sequence of p used.

Table II shows details of the progress of the p-algorithm on Problem 1 and Table III the corresponding progress of the ξ-algorithm.

A bandpass filter of symmetrical structure [10], the first four components of which are a unit element followed by a shunt shorted stub, a series open stub and a shunt shorted stub is considered next, with specifications of 0.1 dB from 1.0875 to 3.2625 GHz (passband) and 50 dB at 0.6 and 3.75 GHz. The initial normalized characteristic impedances were taken as 0.63, 0.33, 1.27, 0.26, 1.27, 0.33, 0.63. The lengths were fixed at normalized values of 1. The functions f_i were set to $\pm$ the difference in dB between the response and specifications, the positive sign corresponding to the passband, the negative sign to the stopband. Twenty-one uniformly spaced passband points were considered but, due to symmetry, only the first 10 were actually used.

We let $\xi=0$, $\eta=0.0001$, $p_1=2$, $p_2=12$, $p_3=72$ and $p_4=432$. 3rd order extrapolation was used leading to 73, 16, 14 and 12 function evaluations per optimization. The sample points were, respectively, 11,11, 7 and 6 for a total of 1149 response evaluations. The final extrapolated solution gives characteristic impedances of 0.606458, 0.303062, 0.722085, 0.235612, 0.722085, 0.303062, 0.606458 (symmetrical as expected). The 8 passband ripples as evaluated at the 21 points fell within 0.06530 - 0.06531 dB. The stopband responses were 50.0347 dB. CPU time on the CDC 6400 was 5 sec. About half this time would be expected if characteristic impedance symmetry were exploited.

The ξ-algorithm generated the results of Table IV employing the actual ripple maxima in the passband as found by quadratic approximations. The lower bound estimation process works whether the specifications are violated or satisfied, yielding an immediate indication of how good a design in the minimax sense one can expect from the results of only one least squares approximation. This table also shows, by comparing columns 3 and 4, that only two optimizations with $p = 2$ of the ξ-algorithm yields, for engineering purposes, essentially equal ripple responses. For a 50 dB specification the final solution has characteristic impedances[a] of 0.606595, 0.303547, 0.722287, 0.235183, 0.722287, 0.303547, 0.606595 with deviations from specifications of -0.028245 (equal to at least 5 figures).

CONCLUSIONS

Two new algorithms and related results for the least pth approach to minimax design have been presented. Documented computer programs are available from the first author at nominal charge. A more detailed presentation of the present material is also available [11].

---

[a] Actually symmetrical to at least the accuracy of the CDC 6400.

REFERENCES

[1] J.W. Bandler and C. Charalambous, "Practical least pth optimization of networks", IEEE Trans. Microwave Theory Tech., vol. MTT-20, Dec. 1972, pp. 834-840.

[2] C. Charalambous and J.W. Bandler, "New algorithms for network optimization", IEEE Trans. Microwave Theory Tech., vol. MTT-21, Dec. 1973, pp. 815-818.

[3] C. Charalambous and J.W. Bandler, "Nonlinear minimax optimization as a sequence of least pth optimization with finite values of p", Int. J. Systems Science, to be published.

[4] J.W. Bandler, C. Charalambous and J.H.K. Chen, "MINOPT - an optimization program based on recent minimax results", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-70, Dec. 1974.

[5] J.W. Bandler and W.Y. Chu, "FLOPT2- a program for least pth optimization with extrapolation to minimax solutions", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-84, Apr. 1975.

[6] C. Charalambous, "Minimax optimization of recursive digital filters using recent minimax results", IEEE Trans. Acoust. Speech and Signal Processing, vol. ASSP-23, Aug. 1975.

[7] A.V. Fiacco and G.P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques. New York: Wiley, 1968.

[8] W.Y. Chu, "Extrapolation in least pth approximation and nonlinear programming", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-71, Dec. 1974.

[9] R. Fletcher, "FORTRAN subroutines for minimization by quasi-Newton methods", Atomic Energy Research Establishment, Harwell, Berkshire, England, Report AERE-R7125, 1972.

[10] M.C. Horton and R.J. Wenzel, "General theory and design of optimum quarter-wave TEM filters", IEEE Trans. Microwave Theory Tech., vol. MTT-13, May 1965, pp. 316-327.

[11] J.W. Bandler, C. Charalambous, J.H.K. Chen and W.Y. Chu, "New results in the least pth approach to minimax design", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-80, Mar. 1975.

TABLE I
OPTIMIZATION OF A THREE-SECTION 10:1 TRANSFORMER OVER 100-PERCENT RELATIVE BANDWIDTH

Effort[b] required to reach or exceed a reflection coefficient of 0.19729 (optimal to 5 figures)

Method	Problem 1				Problem 2		
	Parameter ξ or p	Function Evaluations	Sample Points	Response Evaluations	Function Evaluations	Sample Points	Response Evaluations
ξ-algorithm	0.1	28	11	308	19	11	209
(lower bound)	0.18846	16	7	112	16	7	112
p = 2	0.19730	45	4	180	53	4	212
$\xi^0 = 0$	0.19729	89		600	88		533
p-algorithm	8	39	11	429	29	11	319
3rd order	48	17	8	136	18	8	144
extrapolation	288	15	4	60	14	4	56
$\xi = 0$	1728	12	4	48	11	4	44
$\eta = 0.001$		83		673	72		563
Charalambous -	Alg.1	165	11	1815	105	11	1155
Bandler [2]	Alg. 2	155	11	1705	95	11	1045

[b] Does not include response evaluations to determine sample points to be used

TABLE II
PROGRESS OF THE p-ALGORITHM ON PROBLEM 1

				Max. reflection coefficient	
r	Value of p_r	rth optimum[c]	Extrapolated Solution	rth optimum	at extrapolated solution
1	8	.98828	same	.21017	.21017
		1.62868			
		1.00004			
		3.16228			
		.98828			
		6.13993			
2	48	.99833	1.00035	.19838	.19863
		1.63478	1.63600		
		.99991	.99988		
		3.16228	3.16228		
		.99833	1.00035		
		6.11703	6.11246		
3	288	.99973	1.00000	.19747	.19732
		1.63472	1.63467		
		.99999	1.00000		
		3.16228	3.16228		
		.99973	1.00000		
		6.11726	6.11744		
4	1728	.99995	1.00000	.19732	.19729
		1.63471	1.63471		
		1.00000	1.00000		
		3.16228	3.16228		
		.99995	1.00000		
		6.11730	6.11730		

[c] Parameter vector $[\ell_1/\ell_q\ Z_1\ \ell_2/\ell_q\ Z_2\ \ell_3/\ell_q\ Z_3]^T$ where ℓ_q is the quarter wavelength at center frequency

TABLE III
PROGRESS OF THE ξ-ALGORITHM ON PROBLEM 1

r	Value of ξ^r	rth optimum[c]	Max. reflection coefficient
1	.1	.97238	
		1.59720	
		.98791	.25530
		3.16228	
		.97238	
		6.26097	
2	.18846	.99709	
		1.63451	
		1.00013	.19929
		3.16228	
		.99709	
		6.11804	
3	.19730	1.00000	
		1.63471	
		1.00000	.19729
		3.16228	
		1.00000	
		6.11730	

[c] Parameter vector $[\ell_1/\ell_q\ Z_1\ \ell_2/\ell_q\ Z_2\ \ell_3/\ell_q\ Z_3]^T$ where ℓ_q is the quarter wavelength at center frequency

TABLE IV
LOWER BOUNDS FOR THE SEVEN-SECTION FILTER

Passband Specification 0.1 dB
Value of p_0 2
Value of ξ^0 0

Stopband Specification (dB)	First Maximum Error (dB)	Predicted Lower Bound (dB)	Next Maximum Error (dB)
50	-0.0256	-0.0283	-0.0282
55	0.1430	0.1154	0.1160
60	0.6211	0.4954	0.4986
65	1.5486	1.3148	1.3195

A GENERAL METHOD FOR APPROXIMATION BASED ON THE METHOD OF MULTIPLIERS*

DIMITRI P. BERTSEKAS**
Department of Electrical Engineering and
Coordinated Science Laboratory
University of Illinois, Urbana, Illinois 61801

ABSTRACT

In this paper we consider a method for solving certain optimization problems with constraints, nondifferentiabilities and other ill-conditioning terms in the cost functional by approximating them by well-behaved optimization problems. The approach is based on methods of multipliers. An important characteristic of the approach is that a reduced number of scalar multipliers is utilized over the ordinary multiplier method. Thus, for example, two-sided inequality constraints are handled by means of a single multiplier per constraint as compared with two multipliers per constraint for the ordinary multiplier method. The convergence properties of the methods proposed can be inferred from corresponding properties of multiplier methods with partial elimination of constraints. A related result is provided in this paper.

1. INTRODUCTION

Many optimization problems of interest can be written as

$$\text{minimize} \quad f(x) + \sum_{i=1}^{m} \gamma_i[g_i(x)] \tag{1}$$

$$\text{subject to} \quad x \in X$$

where f, g_i, i = 1, ..., m are continuously differentiable functions in R^n (n-dimensional Euclidean space) and X is a given subset of R^n. The functions $\gamma_i = R \rightarrow (-\infty,+\infty]$ are assumed to be extended real valued, convex and lower semicontinuous convex functions with $\gamma_i(t) < +\infty$ for at least one scalar $t \in R$. We are primarily interested in the case where the presence of the functions γ_i introduce difficulties in the numerical solution of problem (1) in the sence that if the functions γ_i were replaced by some real valued and continuously differentiable functions $\tilde{\gamma}_i$ then problem (1) could be solved in a relatively easy manner. For example the functions γ_i may induce constraints, nondifferentiabilities or ill-conditioning in problem (1). Here are some examples:

Example 1: (Equality Constraints)

$$\gamma_i(t) = \begin{cases} 0 & \text{if } t = 0 \\ +\infty & \text{otherwise} \end{cases} \tag{2}$$

---

*This work was supported in part by the Joint Services Electronics Program under Contract DAAB-07-72-C-0259 and by the National Science Foundation under Grant ENG-76-19332.

**The author is with the Department of Electrical Engineering and the Coordinated Science Laboratory, University of Illinois, Urbana, Illinois 61801.

Here the presence of each $\gamma_i[g_i(x)]$ in problem (1) is equivalent to an additional equality constraint $g_i(x) = 0$.

Example 2: (One-Sided Inequality Constraints)

$$\gamma_i(t) = \begin{cases} 0 & \text{if } t \leq 0 \\ +\infty & \text{otherwise} \end{cases} \tag{3}$$

Here the presence of $\gamma_i[g_i(x)]$ induces the constraint $g_i(x) \leq 0$.

Example 3: (Two-Sided Inequality Constraints)

$$\gamma_i(t) = \begin{cases} 0 & \text{if } \alpha_i \leq t \leq \beta_i \\ +\infty & \text{otherwise} \end{cases} \tag{4}$$

where α_i, β_i are scalars with $\alpha_i < \beta_i$.

Example 4: (Polyhedral Functions)

$$\gamma_i(t) = \max\{0,t\} \tag{5}$$

$$\gamma_i(t) = |t| \tag{6}$$

$$\gamma_i(t) = \begin{cases} |t| & \text{if } |t| \leq \alpha \\ +\infty & \text{otherwise} \end{cases} \tag{7}$$

$$\gamma_i(t) = \begin{cases} \max\limits_{j=1,\dots,r} \{\gamma_j t + \delta_j\} & \text{if } \alpha \leq t \leq \beta \\ +\infty & \end{cases} \tag{8}$$

where α, β, γ_j, δ_j are given scalars.

Example 5: (Ill-Conditioning Terms)

$$\gamma_i(t) = \frac{s}{2} t^2 \tag{9}$$

$$\gamma_i(t) = \alpha e^{\beta t} \tag{10}$$

where s, α, β are given scalars with $s > 0$, $\alpha > 0$. The term (9) may induce ill-conditioning in problem (1) if s is very large while the term (10) may induce ill-conditioning in problem (1) if β is very large. More generally if the second derivatives or third derivatives of γ_i are very large relative to other terms in the cost functional, the numerical solution of problem (1) may run into serious difficulties.

This paper presents an approach for solving numerically problems of the type described above. The approach consists of approximation of problem (1) by a sequence of optimization problems which involve relatively well-behaved objective functions. The approximation is effected by introducing additional variables and constraints in problem (1) thus forming an equivalent constrained minimization problem. This problem is subsequently handled by a suitable method of multipliers (see [1], [2], [14] for analysis and references on multiplier methods). For the case of Examples 1 and 2 our approach turns out to be identical to standard multiplier methods. However for other cases, such as for example the case of two-sided constraints (Example 3), our approach results in computational and storage savings over standard multiplier methods due to the fact that the number of scalar multipliers utilized is significantly reduced. For nearly every case of interest our algorithms may be viewed in effect as multiplier methods with partial elimination of constraints. In the last section of this paper we provide a convergence and rate of convergence result for such methods. This result can be used in turn to provide corresponding convergence results for specific algorithms.

2. THE APPROXIMATION PROCEDURE

It is clear that problem (1) is equivalent to the following problem

$$\text{minimize} \quad f(x) + \sum_{i=1}^{m} \gamma_i [g_i(x) - u_i] \tag{11}$$

$$\text{subject to } x \in X, \quad u_i = 0, \ i=1,\ldots,m$$

where we have introduced the additional variables $u_1,\ldots,u_m$. A method of multipliers for the problem above is based on sequential minimization over $x, u_1,\ldots,u_m$ of the form

$$\text{minimize} \quad f(x) + \sum_{i=1}^{m} \{\gamma_i [g_i(x) - u_i] + y_k^i u_i + \frac{c_k}{2} u_i^2\} \tag{12}$$

$$\text{subject to} \quad x \in X$$

or equivalently

$$\text{minimize} \quad f(x) + \sum_{i=1}^{m} p_{c_k}^i [g_i(x), y_k^i] \tag{13}$$

$$\text{subject to} \quad x \in X$$

where

$$p_{c_k}^i [g_i(x), y_k^i] = \min_{u_i} \{\gamma_i [g_i(x) - u_i] + y_k^i u_i + \frac{c_k}{2} u_i^2\} . \tag{14}$$

The initial multipliers y_o^i, $i=1,\ldots,m$ are arbitrary and after each minimization (13) the multipliers y_k^i are updated by means of

$$y_{k+1}^i = y_k^i + c_k u_i^k \qquad i=1,\ldots,m \tag{15}$$

where u_i^k, $i=1,\ldots,m$ solve (13) (together with some vector x_k). Alternate methods could be obtained by using a nonquadratic penalty function in (12) and in fact in several cases such nonquadratic penalty functions are desirable. In order to keep the exposition simple we will restrict ourselves for the moment to quadratic penalty functions.

It is important to note that the function $p^i_{c_k}$ of (14) is both real valued and continuously differentiable in x provided the function g_i is continuously differentiable and hence problem (13) can be solved by the powerful methods available for differentiable functions whenever f, g_i are differentiable. These properties of the function $p^i_{c_k}$ can be inferred from the following lemma:

Lemma 1: Let $\gamma : R \to (-\infty, +\infty]$ be a lower semicontinuous convex function and assume $\gamma(t) < +\infty$ for at least one scalar $t \in R$. Let also λ and $c > 0$ be two scalars. Then the function $p_c(\cdot,\lambda)$ defined by

$$p_c(t,\lambda) = \inf_u \{\gamma(t-u) + \lambda u + \frac{c}{2} u^2\} \tag{16}$$

is real valued, convex and continuously differentiable in t. Furthermore the infimum with respect to u in (16) is attained at a unique point for every $t \in R$.

Proof: The function $p_c(\cdot,t)$ is the infimal convolution [13] of the convex function γ and the quadratic convex function $r: R \to R$ defined by

$$r(u) = \lambda u + \frac{c}{2} u^2. \tag{17}$$

Since $r(u) \to \infty$ as $|u| \to \infty$ it follows from Corollary 9.2.2 of [13] that $p_c(\cdot,\lambda)$ is convex and the infimum in (16) is attained for each t by some u. Since r is strictly convex real valued it follows that $p_c(\cdot,\lambda)$ is also real valued and the minimum is attained at a single point. Also r is a smooth function ([13], p. 251), and from Corollary 26.3.2 of [13] it follows that $p_c(\cdot,\lambda)$ is an essentially smooth convex function. Since it is also real valued, it is continuously differentiable.

Q. E. D.

The interpretation of $p_c(\cdot,\lambda)$ in the proof above as the infimal convolution of γ and r defined by (17) is useful in visualizing the form of $p_c(\cdot,\lambda)$. The epigraph of $p_c(\cdot,\lambda)$ is obtained as the vector sum of the epigraph of the functions $\gamma(\cdot)$ and $r(\cdot)$ (see [13] Theorem 5.4).

It is to be noted that even though we employed the additional variables $u_1,\dots,u_m$ in order to introduce the algorithm, the numerical computation itself need not involve these variables since in the cases of interest to us the functions $p^i_{c_k}$ of (13), (14) can be obtained in explicit form. Furthermore the minimizing scalars u^k_i of (15) can be expressed directly in terms of minimizing vectors x_k in problem (13) since u^k_i is uniquely defined in terms of x_k, c_k, y^i_k as the minimizing scalar in (14). We provide some examples below:

Example 1: For the case where γ_i is given by (2), ($\gamma_i(0) = 0$, $\gamma_i(t) = +\infty$ if $t \neq 0$) we obtain from (14), (15)

$$p^i_{c_k}[g_i(x), y^i_k] = y^i_k g_i(x) + \frac{c_k}{2}[g_i(x)]^2$$

$$y^i_{k+1} = y^i_k + c_k g_i(x_k) \qquad i=1,\dots,m .$$

In this case the iteration reduces to the ordinary multiplier iteration with quadratic penalty function as proposed for equality constraints by Hestenes [9].

Example 2: For γ_i given by (3) ($\gamma_i(t) = 0$ if $t \leq 0$, $\gamma_i(t) = +\infty$ if $0 < t$) we obtain from (14), (15) by straightforward calculation

$$p^i_{c_k}[g_i(x),y^i_k] = \frac{1}{2c_k}\{(\max[0,y^i_k + c_k g_i(x)])^2 - (y^i_k)^2\}$$

$$y^i_{k+1} = \max[0,y^i_k + c_k g_i(x_k)] \qquad i=1,\ldots,m \;.$$

The algorithm reduces to the multiplier method proposed for inequality constraints by Rockafellar [12].

Example 3: For the case of two-sided inequality constraints ($\gamma_i(t) = 0$ if $\alpha_i \leq t \leq \beta_i$, $\gamma_i(t) = +\infty$ $t \notin [\alpha_i,\beta_i]$ c.f. (4)) we obtain

$$p^i_{c_k}[g_i(x),y^i_k] = \begin{cases} y^i_k[g_i(x)-\beta_i] + \frac{c_k}{2}[g_i(x)-\beta_i]^2 & \text{if } \beta_i + \frac{y^i_k}{c_k} \leq g_i(x) \\ y^i_k[g_i(x)-\alpha_i] + \frac{c_k}{2}[g_i(x)-\alpha_i]^2 & \text{if } g_i(x) \leq \alpha_i + \frac{y^i_k}{c_k} \\ -\frac{(y^i_k)^2}{2c_k} & \text{otherwise} \end{cases}$$

$$y^i_{k+1} = \begin{cases} y^i_k + c_k[g_i(x_k)-\beta_i] & \text{if } \beta_i + \frac{y^i_k}{c_k} \leq g_i(x_k) \\ y^i_k + c_k[g_i(x_k)-\alpha_i] & \text{if } g_i(x_k) \leq \alpha_i + \frac{y^i_k}{c_k} \\ 0 & \text{otherwise} \end{cases}$$

Notice that this algorithm involves a single multiplier per two-sided constraint as compared with two multipliers per constraint required by the ordinary multiplier method. For further analysis concerning this algorithm we refer to [8].

Example 4: Consider the case where $\gamma_i(t) = \max[0,t]$. Then by straightforward calculation we obtain

$$p_{c_k}^i[g_i(x),y_k^i] = \begin{cases} g_i(x) - \frac{(1-y_k^i)^2}{2c_k} & \text{if } g_i(x) \geq \frac{1-y_k^i}{c_k} \\ -\frac{(y_k^i)^2}{2c_k} & \text{if } g_i(x) \leq -\frac{y_k^i}{c_k} \\ y_k^i g_i(x) + \frac{c_k}{2}[g_i(x)]^2 & \text{if } -\frac{y_k^i}{c_k} \leq g_i(x) \leq \frac{1-y_k^i}{c_k} \end{cases}$$

$$y_{k+1}^i = \begin{cases} 1 & \text{if } g_i(x_k) \geq \frac{1-y_k^i}{c_k} \\ 0 & \text{if } g_i(x_k) \leq -\frac{y_k^i}{c_k} \\ y_k^i + c_k g_i(x_k) & \text{if } -\frac{y_k^i}{c_k} \leq g_i(x_k) \leq \frac{1-y_k^i}{c_k}\ . \end{cases}$$

The corresponding algorithm is a special case of an approximation method for nondifferentiable optimization given in [6]. Notice that a single multiplier per term $\gamma_i[g_i(x)]$ is utilized. If one were to convert the problem to a nonlinear programming problem of the form

$$\text{minimize} \quad f(x) + \sum_{i=1}^{m} z_i$$

$$\text{subject to} \quad g_i(x) \leq z_i, \quad 0 \leq z_i, \quad i = 1,\dots,m$$

where z_i are additional variables, then two multipliers per term $\gamma_i[g_i(x)]$ would be required in order for the problem to be solved by the method of multipliers.

The case where $\gamma_i(t) = |t|$ can be converted to the earlier case by writing $|t| = -t + \max[0,2t]$.

Let $\gamma_i(t) = |t|$ for $|t| \leq \alpha$ and $\gamma_i(t) = +\infty$ for $|t| > \alpha$ where α is some positive number. Such terms appear for example in the cost functional of minimum fuel problems. We have by straightforward calculation

$$p^i_{c_k}[g_i(x),y^i_k] = \begin{cases} \alpha + y^i_k[g_i(x)+\alpha] + \frac{c_k}{2}[g_i(x)+\alpha]^2 & \text{if } g_i(x) \le -\alpha - \frac{1+y^i_k}{c_k} \\ -g_i(x) - \frac{(1+y^i_k)^2}{2c_k} & \text{if } -\alpha - \frac{1+y^i_k}{c_k} \le g_i(x) \le -\frac{1+y^i_k}{c_k} \\ y^i_k g_i(x) + \frac{c_k}{2}[g_i(x)]^2 & \text{if } -\frac{1+y^i_k}{c_k} \le g_i(x) \le \frac{1-y^i_k}{c_k} \\ g_i(x) - \frac{(1-y^i_k)^2}{2c_k} & \text{if } \frac{1-y^i_k}{c_k} \le g_i(x) \le \alpha + \frac{1-y^i_k}{c_k} \\ \alpha + y^i_k[g_i(x)-\alpha] + \frac{c_k}{2}[g_i(x)-\alpha]^2 & \text{if } \alpha + \frac{1-y^i_k}{c_k} \le g_i(x) \end{cases}$$

and the iteration (15) takes the form

$$y^i_{k+1} = \begin{cases} y^i_k + c_k[g_i(x_k) + \alpha] & \text{if } g_i(x_k) \le -\alpha - \frac{1+y^i_k}{c_k} \\ -1 & \text{if } -\alpha - \frac{1+y^i_k}{c_k} \le g_i(x_k) \le -\frac{1+y^i_k}{c_k} \\ y^i_k + c_k g_i(x) & \text{if } -\frac{1+y^i_k}{c_k} \le g_i(x_k) \le \frac{1-y^i_k}{c_k} \\ 1 & \text{if } \frac{1-y^i_k}{c_k} \le g_i(x_k) \le \alpha + \frac{1-y^i_k}{c_k} \\ y^i_k + c_k[g_i(x_k) - \alpha] & \text{if } \alpha + \frac{1-y^i_k}{c_k} \le g_i(x_k) \end{cases}$$

Notice that a single multiplier per term γ_i is utilized in place of four multipliers per term γ_i for the ordinary method of multipliers.

Similarly one may obtain the function $p^i_{c_k}$ and the iteration (15) in explicit form for the function γ_i of (8). Again only one multiplier per term is required in place of r multipliers for the ordinary multiplier method.

Example 5: Let $\gamma_i(t) = \frac{s_i}{2} t^2$ (c.f. (9)). Then we have by straightforward calculation

$$p_{c_k}^i [g_i(x), y_k^i] = \frac{s_i}{s_i + c_k} \{ \frac{c_k}{2} [g_i(x)]^2 + y_k^i g_i(x) - \frac{(y_k^i)^2}{2 s_i} \}$$

and the iteration (15) takes the form

$$y_{k+1}^i = y_k^i + c_k \frac{s_i g_i(x_k) - y_k^i}{s_i + c_k} .$$

Notice that the second derivative of $p_{c_k}^i(\cdot, y_k^i)$ given above is $\frac{s_i c_k}{s_i + c_k}$ and can be made arbitrarily small by choosing c_k sufficiently small.

A slightly different type of algorithm may be obtained when the functions γ_i in problem (1) are monotonically nondecreasing. Then we may consider the following problem which is equivalent to problem (1)

$$\text{minimize} \quad f(x) + \sum_{i=1}^{m} \gamma_i [g_i(x) - u_i] \tag{18}$$

$$\text{subject to} \quad x \epsilon X, \ u_i \leq 0 , \quad i=1,\dots,m .$$

and we may utilize a multiplier method for solving the problem above. An example is provided for the case where $\gamma_i(t) = \alpha e^{\beta t}$ with $\alpha, \beta > 0$ as in (10). For this case it is convenient to use a multiplier method with exponential penalty function proposed in 1972 by B. Kort and the author [10]. The penalty function for this method is $\frac{1}{c} y e^{cu}$ and we have

$$p_{c_k}^i [g_i(x), y_k^i] = \min_u \{ \alpha e^{\beta(t-u)} + \frac{1}{c_k} y_k^i e^{c_k u} \} =$$

$$= (\alpha \beta)^{\frac{c_k}{c_k + \beta}} (y_k^i)^{\frac{\beta}{c_k + \beta}} \frac{c_k + \beta}{c_k \beta} e^{\frac{c_k \beta}{c_k + \beta} g_i(x)}$$

The multiplier iteration is $y_{k+1}^i = y_k^i e^{c_k u_k}$ which yields

$$y_{k+1}^i = (y_k^i)^{\frac{\beta}{c_k + \beta}} (\alpha \beta)^{\frac{c_k}{c_k + \beta}} e^{\frac{c_k \beta}{c_k + \beta} g_i(x_k)}$$

An application of this method is given elsewhere in these proceedings [7] where an algorithm is described for solving electric network problems involving physical or ideal diodes. The voltage-current characteristics of physical diodes are typically described by sharply rising exponentials which cause formidable difficulties during computational solution of the related network problem. By using the multiplier method with exponential penalty function described above all computational difficulties due to steep diode characteristics are bypassed.

3. CONVERGENCE ANALYSIS - MULTIPLIER METHODS WITH PARTIAL ELIMINATION OF CONSTRAINTS

From the discussion given above it should be evident that algorithms of the type that we have discussed are multiplier methods where only some of the constraints (i.e. the constraints $u_i = 0$) are eliminated by means of a penalty function. For example in the problem involving two-sided inequality constraints

$$\text{minimize} \quad f(x) \tag{19}$$

$$\text{subject to} \quad x \epsilon X, \ \alpha_i \leq g_i(x) - u_i \leq \beta_i, \quad u_i = 0, \quad i=1,\ldots,m$$

only the constraints $u_i = 0$ are eliminated by means of a penalty function. The approximate minimization problem takes the form

$$\text{minimize} \quad f(x) + \sum_{i=1}^{m} (y_k^i u_i + \frac{c_k}{2} u_i^2)$$

$$\text{subject to} \quad x \epsilon X, \ \alpha_i \leq g_i(x) - u_i \leq \beta_i, \ i=1,\ldots,m$$

This problem may also be converted to the following problem involving equality constraints by introducing additional variables z_i, w_i, $i=1,\ldots,m$

$$\text{minimize} \quad f(x) + \sum_{i=1}^{m} (y_k^i u_i + \frac{c_k}{2} u_i^2) \tag{20}$$

$$\text{subject to } x \epsilon X, \ \alpha_i + z_i^2 = g_i(x) - u_i, \ g_i(x) - u_i + w_i^2 = \beta_i, \ i=1,\ldots,m.$$

Thus we may restrict ourselves to the case of equality constraints.

In what follows in this section we consider an optimization problem involving equality constraints and a multiplier algirithm for its solution involving partial elimination of constraints. A convergence and rate of convergence result for this algorithm will be given. This result yields in a straightforward manner convergence and rate of convergence results for algorithms such as those considered in the previous section.

Consider the problem

$$\text{minimize} \quad q(x) \tag{21}$$

$$\text{subject to} \quad h_i(x) = 0, \quad \ell_j(x) = 0, \quad i=1,\ldots,m, \; j=1,\ldots,r.$$

Let $\bar{x}$ be a local minimum for this problem. We assume that the functions $q:R^n \to R$, $h_i:R^n \to R$, $\ell_j:R^n \to R$ have Hessian matrices $\nabla^2 q(x)$, $\nabla^2 h_i(x)$, $\nabla^2 \ell_j(x)$ which are Lipschitz continuous in a neighborhood of $\bar{x}$. Furthermore we assume that the gradients $\nabla h_i(\bar{x})$, $\nabla \ell_j(\bar{x})$, $i=1,\ldots,m$, $j=1,\ldots,r$ are linearly independent. As a consequence we obtain that there exist unique Lagrange multipliers $\bar{\lambda} = (\bar{\lambda}^1,\ldots,\bar{\lambda}^m)'$, $\bar{\mu} = (\bar{\mu}^1,\ldots,\bar{\mu}^r)'$ such that

$$\nabla q(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}^i \nabla h_i(\bar{x}) + \sum_{j=1}^{r} \bar{\mu}^j \nabla \ell_j(\bar{x}) = 0. \tag{22}$$

Consider now the following multiplier iteration for solving problem (21)

$$\lambda_{k+1}^i = \lambda_k^i + c_k h_i(x_k) \qquad i=1,\ldots,m \tag{23}$$

where x_k solves, within a neighborhood of $\bar{x}$, the problem

$$\text{minimize} \quad q(x) + \sum_{i=1}^{m} \{\lambda_k^i h_i(x) + \frac{c_k}{2} [h_i(x)]^2\} \tag{24}$$

$$\text{subject to} \quad \ell_j(x) = 0 \quad j=1,\ldots,r.$$

In the above relations $\{c_k\}$ is a sequence of positive penalty parameters and $\lambda_o = (\lambda_o^1,\ldots,\lambda_o^m)'$ is a given initial multiplier vector.

The convergence properties of the multiplier method specified by (23), (24) may be derived by casting this method into the usual form of the method of multipliers where all constraints are eliminated by means of a penalty function, by using the constraints $\ell_j(x) = 0$ to reduce the number of the variables of the problem in a manner which is familiar from constructions used in the reduced gradient method (see e.g. [11]). By utilizing a result proved for such multiplier methods [2]-[4] we may obtain the following proposition (see [8] for the detailed proof).

Proposition 1: In addition to linear independence of the gradients $\nabla h_i(\bar{x})$, $\nabla \ell_j(\bar{x})$, $i=1,\ldots,m$, $j=1,\ldots,r$ and Lipschitz continuity of the Hessian matrices $\nabla^2 q(x)$, $\nabla^2 h_i(x)$, $\nabla \ell_j(x)$, $i=1,\ldots,m$, $j=1,\ldots,r$ within a neighborhood of $\bar{x}$, assume that

$$w'\{\nabla^2 q(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}^i \nabla^2 h_i(\bar{x}) + \sum_{j=1}^{r} \bar{\mu}^j \nabla^2 \ell_j(\bar{x})\} w > 0$$

for all $w \in R^n$ such that $w \neq 0$ and $w' \nabla h_i(\bar{x}) = 0$, $w' \nabla \ell_j(\bar{x}) = 0$ for all $i=1,\ldots,m$, $j=1,\ldots,r$.

Then for any subset $\Lambda \subset R^m$ there exists a scalar $c^* > 0$ (depending on Λ) such that for all $c \geq c^*$ and $\lambda = (\lambda^1,\ldots,\lambda^m)' \epsilon \Lambda$ the problem

$$\text{minimize} \quad q(x) + \sum_{i=1}^{m} \{\lambda^i h_i(x) + \frac{c}{2} [h_i(x)]^2\}$$

$$\text{subject to} \quad \ell_j(x) = 0 \quad j=1,\ldots,r$$

has a unique solution within some neighborhood of $\bar{x}$ denoted by $x(\lambda,c)$. Furthermore there exists a scalar $M > 0$ such that for all $c \geq c^*$ and $\lambda \epsilon \Lambda$

$$\|x(\lambda,c)-\bar{x}\| \leq \frac{M\|\lambda-\bar{\lambda}\|}{c}$$

$$\|\tilde{\lambda}(\lambda,c)-\bar{\lambda}\| \leq \frac{M\ \lambda-\bar{\lambda}\|}{c}$$

where $\bar{\lambda}$ is the Lagrange multiplier vector for problem (21) (cf. (22)) and the vector $\tilde{\lambda}(\lambda,c)$ has coordinates given by

$$\tilde{\lambda}^i(\lambda,c) = \lambda^i + ch_i[x(\lambda,c)] \quad i=1,\ldots,m$$

This proposition yields easily a convergence and rate of convergence result for the multiplier method (23), (24) which we state in the following proposition:

Proposition 2: Under the assumptions of Proposition 1 let Λ be an arbitrary sphere centered at the vector $\bar{\lambda}$. Consider also a non-decreasing penalty parameter $\{c_k\}$ such that for some integer $\bar{k} \geq 0$ we have $c_{\bar{k}} > \max\{M,c^*\}$ where M and c^* correspond to Λ as in Proposition 1. Let $\{\lambda_k\}$ be a sequence such that $\lambda_{\bar{k}} \epsilon \Lambda$ and for all $k \geq \bar{k}$

$$\lambda_{k+1} = \lambda_k + c_k h[x(\lambda_k,c_k)] .$$

Then $x(\lambda_k,c_k) \to \bar{x}$, $\lambda_k \to \bar{\lambda}$. Furthermore if $c_k \to \infty$ the convergence of $\{\|\lambda_k - \bar{\lambda}\|\}$ to zero is superlinear.

REFERENCES

[1] Bertsekas, D. P., "Combined Primal-Dual and Penalty Methods for Constrained Minimization," SIAM J. on Control, Vol. 13, 1975, pp. 521-544.

[2] Bertsekas, D. P., "Multiplier Methods: A Survey," Preprints of IFAC 6th Triennial World Congress, Part IB, Boston, Mass., August 1975; also Automatica, March 1976 (to appear).

[3] Bertsekas, D. P., "On Penalty and Multiplier Methods for Constrained Minimization," in Nonlinear Programming II, O. Mangasarian, R. Meyer, S. Robinson (eds.), Academic Press, New York, 1975.

[4] Bertsekas, D. P., "On Penalty and Multiplier Methods," Dept. of Engrg.-Economic Systems Working Paper, Stanford Univ., Stanford, Calif., 1973, to appear in SIAM J. on Control, Feb. 1976.

[5] Bertsekas, D. P., "Convergence Rate of Penalty and Multiplier Methods," Proc. 1973 IEEE Conf. on Decision and Control, San Diego, Calif., pp. 260-264.

[6] Bertsekas, D. P., "Nondifferentiable Optimization via Approximation," Proc. of Twelfth Annual Allerton Conf. on Circuit and System Theory, Allerton Park, Ill., Oct. 1974, pp. 41-52, also in Mathematical Programming Study 3, Balinski, M. and Wolfe, P. (eds.), North Holland, 1975 (to appear).

[7] Bertsekas, D. P., "A New Algorithm for Analysis of Nonlinear Resistive Networks," Proc. Thirteenth Annual Allerton Conf. on Circuit and System Theory, Allerton Park, Ill., Oct. 1975.

[8] Bertsekas, D. P., "Multiplier Methods for Two-Sided Inequality Constraints and Related Algorithms," Coordinated Science Laboratory Working Paper, Univ. of Illinois, Aug. 1975.

[9] Hestenes, M. R., "Multiplier and Gradient Methods," J. Optimization Theory Appl., Vol. 4, 1969, pp. 303-320.

[10] Kort, B. W. and D. P. Bertsekas, "A New Penalty Function Algorithm for Constrained Minimization," Proc. 1972 IEEE Conference on Decision and Control, New Orleans, Louisiana, Dec. 1972.

[11] Luenberger, D. C., Introduction to Linear and Nonlinear Programming, Addison-Wesley, Reading, Mass., 1973.

[12] Rockafellar, R. T., "The Multiplier Method of Hestenes and Powell Applied to Convex Programming," J. Optimization Theory and Applications, Vol. 12, 1973, pp. 555-562.

[13] Rockafellar, R. T., Convex Analysis, Princeton Univ. Press, Princeton, N. J., 1970.

[14] Rockafellar, R. T., "Solving a Nonlinear Programming Problem by Way of a Dual Problem," Symposia Mathematica, to appear.

A MODIFIED SIGN DETECTOR BASED ON A FOUR-LEVEL INPUT QUANTIZATION

SALEEM A. KASSAM
Dept. of Systems Eng.
Moore School of Elec. Eng.
Univ. of Pennsylvania
Philadelphia, PA 19174

and

JOHN B. THOMAS
Dept. of Electrical Eng.
Princeton University
Princeton, NJ 08540

ABSTRACT

A simple modification of the sign detector is considered for nonparametric detection under a hypothesis of symmetry of the noise densities. The modified detector utilizes four levels of quantization and retains a constant false alarm probability by employing a conditional test. The modified detector remains simple to implement and exhibits better performance than the simple sign detector, as shown both by asymptotic and computed-performance comparisons.

I. INTRODUCTION

For the detection of a known signal in additive noise the sign detector [1] results in a simple system which has a constant false alarm probability for the class of noise densities with zero medians. The performance of the sign detector is often quite poor relative to the optimum detectors for specific noise statistics. In the case of Gaussian noise, for example, the optimum correlator performs much better; the asymptotic relative efficiency (ARE) of the sign detector relative to the correlator for Gaussian noise is only 0.64. More complex nonparametric detectors, such as those based on ranks, usually exhibit much better performance than does the sign detector. The major asset of the sign detector lies in its extreme simplicity of implementation.

Both the simplicity and nonparametric property of the sign detector are a consequence of the hard-limiting, or two-level quantization, performed on the input data. It is also this hard-limiting which results in the performance being generally poor as compared to other detectors such as the correlator. A logical development is to increase the number of levels of quantization for better performance, while maintaining this number at a small enough value to allow simple implementation. Detectors which are based essentially on this idea have been considered in [2,3]. To retain nonparametric operation with such higher-level quantization, these detectors [2,3] are usually based on some further specific assumptions about the noise distributions.

In this paper we introduce a simple modified sign detector which is based on a symmetric four-level quantization of the input data. We will show how, using a conditional test, a constant false alarm probability can be maintained under only the mild general assumption of symmetry of the noise density functions. For simplicity, we consider only the detection of a constant, positive signal in additive noise. Extensions to the more general case of a know signal and the modification of other detectors (such as the polarity coincidence correlator) can be obtained in a similar manner.

II. THE NONPARAMETRIC FOUR-LEVEL SIGN DETECTOR

Let $\underline{X} = (X_1, X_2, \ldots, X_n)$ be the observed n-vector of independent samples, each with probability density function $p(x) = f(x-s)$. The constant s represents a signal amplitude, the hypothesis H to be tested being that s=0 and the alternative K being s>0, with the assumption on f(x) that

$$f(x) = f(-x). \tag{1}$$

Thus, the hypothesis H is that of symmetry with respect to the origin of the noise density function.

Let the nonlinearity $h(x;a,\ell)$ defined as in Fig. 1 be the characteristic of a symmetric four-level quantizer. For a=0 or ℓ=1 we get the hard limiter of the sign detector. We now consider detection based on the statistic

$$S_{a,\ell}(\underline{X})=\sum_{i=1}^{n} h(x_i;a,\ell) \tag{2}$$

which, of course, reduces to the sign test statistic for a=0 or ℓ=1.

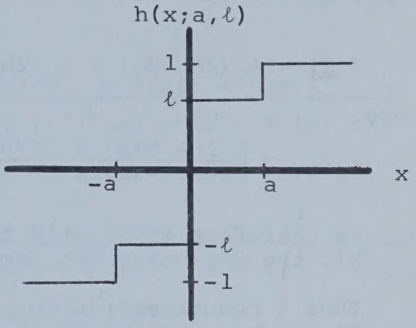


Fig.1 The Nonlinearity $h(x;a,\ell)$

The distribution of $S_{a,\ell}(\underline{X})$ even under the hypothesis H depends on the density function f(x). More specifically, this distribution depends on F(a), where F(x) is the distribution function corresponding to the density f(x). Thus a fixed threshold test based on $S_{a,\ell}(\underline{X})$ cannot have a constant false alarm probability for all noise densities in H. Let us consider a detector which reduces the input data vector $\underline{X}$ to four statistics N_1, N_2, N_{-1} and N_{-2}, where

$$\begin{aligned} N_1 &= \text{\# of samples in } [a,\infty] \\ N_2 &= \text{\# of samples in } [0,a] \\ N_{-2} &= \text{\# of samples in } [-a,0] \\ N_{-1} &= \text{\# of samples in } [-\infty,-a] \end{aligned} \tag{3}$$

Define $M_1 = N_1 + N_{-1}$ and $M_2 = N_2 + N_{-2}$. Now M_1 is a statistic which is complete [4],and sufficient [4] for the N_1, N_2, N_{-1} and N_{-2}, under H. Completeness follows from the definition of the class of density functions comprising H, and sufficiency follows from the symmetry condition of Eq. (1); given M_1, the distribution of N_1 is binomial with parameters $(1/2, M_1)$, under H. From the Lehmann-Scheffe Theorem [4, p.87], we have that the best similar size-α test for H based on the statistics N_1, N_2, N_{-1} and N_{-2} is a test conditioned on M_1. A similar size-α test gives a detector with a constant false alarm probability α for all measures of H. We then simply apply the Neyman-Pearson lemma to get the locally best (conditional) test [5] for a specific shift alternative. The result is easy to obtain, because the conditional distributions involved are all binomial. We find that the locally best test, which is a test conditioned on M_1, uses a test statistic $T(\underline{X})$ given by

$$T(\underline{X}) = \ell_1(2N_1-M_1) + \ell_2(2N_2-M_2), \tag{4}$$

where

$$\ell_2/\ell_1 = \left[\frac{1 - F(a)}{F(a) - 1/2}\right]\left[\frac{f(0) - f(a)}{f(a)}\right] . \tag{5}$$

$T(\underline{X})$ is therefore of exactly the same form as the test statistic $S_{a,\ell}(\underline{X})$, the two being the same if $\ell = \ell_2/\ell_1$.

Thus a nonparametric detector using the test statistic of Eq. (2) can be obtained by incorporating a conditional structure in the detector. In addition to computing $S_{a,\ell}(\underline{X})$, the four-level sign detector also obtains the statistic M_1. For each realization m_1 of M_1 the values of the threshold $t(m_1)$ and the randomization $r(m_1)$ [for the case when $S_{a,\ell}(\underline{X}) = t(m_1)$] are obtained to give the desired false alarm probability α conditioned on $M_1 = m_1$. Thus the detector can have a constant false alarm probability α for the whole hypothesis H. The n+1 possible values of the threshold and randomization can be computed and stored easily in a read-only memory device. These threshold and randomization values are independent of the parameter a in Fig. 1. The structure of the resulting modified four-level sign detector is shown in Fig. 2.

For the case where $\ell=0$, we get a particularly simple (three-level) sign detector. This may also be called the dead-zone limiter detector, because of the shape of $h(x;a,0)$. This dead-zone limiter detector also exhibits very good performance compared to the simple sign detector. By considering even higher levels of quantization, conditionally nonparametric detectors with better performance can be obtained.

III. PERFORMANCE

To compare the performances of two detectors we may use their ARE, which gives an analytic expression for the relative efficiency in the asymptotic case. This asymptotic performance comparison gives a good indication of performance in the finite sample case in the present case, as we will see. The ARE of two detectors may in most cases be obtained as a ratio of their efficacies [4]. The efficacy of a detector is defined as an incremental signal-to-noise ratio, and can be obtained easily for the detectors in which we are interested. For the case of conditional tests, justification and validity of ARE comparisons must be established, and come from two theorems due to Hoeffding, which are discussed in [4, Ch. 7]. For a complete derivation of the following equations see [6].

The ARE of the four-level sign detector relative to the simple sign detector for testing H vs. K is given by

$$ARE_{4s,2s}(f;a,\ell) = \frac{2\{\ell[f(0) - f(a)] + f(a)\}^2}{4f^2(0)\,\{\ell^2[F(a) - 1/2] + 1-F(a)\}}, \quad (6)$$

and the ARE relative to the linear detector is given by

$$ARE_{4s,L}(f;a,\ell) = \frac{2\sigma^2\{\ell[f(0) - f(a)] + f(a)\}^2}{\ell^2[F(a) - 1/2] + [1-F(a)]} \quad (7)$$

where σ^2 is the variance of the density f(x).

For a given f(x) and parameter a these ARE's can be maximized with respect to ℓ. The optimum value of ℓ is given by the ratio ℓ_2/ℓ_1 of Eq. (5), the resulting maximum values of the ARE's being

$$ARE_{4s,2s}(f;a) = \frac{2f^2(a)}{4f^2(0)[1-F(a)]} + \frac{2[f(0) - f(a)]^2}{4f^2(0)[F(a) - 1/2]} \quad (8)$$

and

$$ARE_{4s,L}(f;a) = \frac{2\sigma^2 f^2(a)}{[1-F(a)]} + \frac{2\sigma^2[f(0) - f(a)]^2}{F(a) - 1/2} . \quad (9)$$

The parameter a can also be picked optimally for a given density function. For the Gaussian case, the optimum value a* of a and the corresponding optimum value ℓ^* of ℓ are $a^* = 0.98\sigma$ and $\ell^* = 0.3$, giving an $ARE_{4s,L}$ of 0.88 and an $ARE_{4s,2s}$ of 1.38.

Thus a significant performance improvement over the simple sign detector is obtained by the four-level nonparametric sign detector. It can also be shown that the performance of the four-level sign detector is not very sensitive to deviations of the values of a and ℓ from the optimum values; and that the detector can be designed to obtain good performance for large classes of symmetric noise densities.

Since the computation of the exact power for specific alternatives involves binomial distributions only, the computations can be programmed to be done on a digital computer. For a given sample size n, false alarm probability α, and level ℓ the tables of thresholds and randomization probabilities are first obtained. The detection power can then be computed for any value of the signal s and parameter a. The results are shown in Table I, for the case of Gaussian noise. Also shown in Table I are the powers of the sign and linear detectors for comparison. These results bear out the previous conclusions based on ARE results.

For the case $\ell = 0$ which gives the dead-zone limiter detector, the ARE's follow directly from Eqs. (6) and (7). For the optimum choice of a $(a = 0.61\sigma)$ in the case of Gaussian noise, an $ARE_{4s,L}$ of 0.81 can be obtained. Thus even the three-level dead-zone limiter detector has significantly better performance than the simple sign detector. Once again finite-sample power may be computed. The results for a particular choice of n and α, with the parameter a being chosen to be optimum, are shown in Table II for Gaussian noise.

IV. CONCLUDING REMARKS

We have seen that a simple extension of the sign detector based on a conditional test and higher-level quantization leads to a useful, more efficient nonparametric detector. The increased efficiency was established through both asymptotic performance comparisons and finite-sample power computations. Only the simple case of detection of a constant signal was considered; the extension to the case of a known signal may also be easily obtained. The optimal setting of the parameter a of the quantizer considered in this paper, possibly by a recursive algorithm, remains an unexplored aspect of the detection system.

ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation under Grant GK-24187 and by the U.S. Army Research Office under contract DAHC04- 75-G-0192

REFERENCES

1. J.W. Carlyle, "Nonparametric methods in detection theory", in Communication Theory, A.V. Balakrishnan, Ed. New York: McGraw-Hill, 1968, Ch.8.

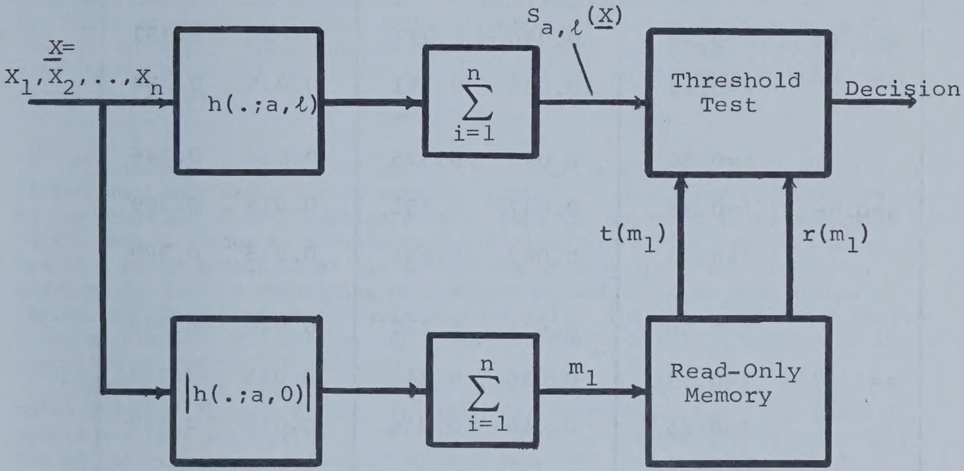
2. Y.C. Ching and L. Kurz, "Nonparametric detectors based on m-interval partitioning", IEEE Trans.on Inform. Theory, Vol. IT-18, pp. 251-257, March 1972.

3. D.G. Lainiotis, "On a class of generalized polarity coincidence correlators", IEEE Trans. on Inform. Theory, Vol. IT-14, pp. 603-604, July 1968.

4. D.A.S. Fraser, Nonparametric Methods in Statistics. New York: John Wiley, 1957.

5. T.S. Ferguson, Mathematical Statistics. New York: Academic Press, 1967.

6. S.A. Kassam, "Nonparametric and Robust Detection: Dependent Data and Conditional Tests", Ph.D. dissertation, Dept. Elec. Eng., Princeton University, Princeton, NJ, August 1975.



$\underline{X}=$
$X_1, X_2, ..., X_n$
$h(.;a,\ell)$
$\sum_{i=1}^{n}$
$S_{a,\ell}(\underline{X})$
Threshold Test
Decision
$t(m_1)$
$r(m_1)$
$|h(.;a,0)|$
$\sum_{i=1}^{n}$
m_1
Read-Only Memory

Fig.2 Structure of the Modified Sign Detector

TABLE I. DETECTION POWER FOR THE MODIFIED SIGN DETECTOR, THE SIGN DETECTOR, AND THE LINEAR DETECTOR; GAUSSIAN NOISE WITH UNIT VARIANCE, CONSTANT ADDITIVE SIGNAL s .

		$\alpha=0.01$, $n=50$		$\alpha=0.001$, $n=100$	
		$s=0.1$	$s=0.3$	$s=0.1$	$s=0.3$
	$\ell=0.50$	0.045	0.325	0.014	0.331
$a=0.60$	$\ell=0.33$	0.046	0.339	0.014	0.351
	$\ell=0.25$	0.046	0.341	0.015	0.354
	$\ell=0.50$	0.045	0.335	0.014	0.346
$a=0.98$	$\ell=0.33$	0.047	0.351	0.015	0.369
	$\ell=0.25$	0.047	0.351	0.015	0.369
	$\ell=0.50$	0.045	0.330	0.014	0.340
$a=1.20$	$\ell=0.33$	0.046	0.346	0.015	0.363
	$\ell=0.25$	0.046	0.344	0.015	0.359
Sign Detector		0.038	0.252	0.011	0.232
Linear Detector		0.053	0.419	0.018	0.464

TABLE II. DETECTION POWER FOR MODIFIED SIGN DETECTOR WITH $\ell=0$; GAUSSIAN NOISE WITH UNIT VARIANCE.

	$\alpha=0.01$, $n=50$		$\alpha=0.001$, $n=100$	
parameter a	$s=0.1$	$s=0.3$	$s=0.1$	$s=0.3$
0.60	0.044	0.320	0.014	0.326
0.98	0.041	0.285	0.012	0.281
1.20	0.037	0.242	0.010	0.224

LOCALLY OPTIMUM RECEIVER STRUCTURES FOR KNOWN SIGNALS IN NON-GAUSSIAN NARROWBAND NOISE[†]

J. W. MODESTINO
Electrical and Systems Engineering Department
Rensselaer Polytechnic Institute, Troy, New York

ABSTRACT

The concept of a locally optimum receiver structure is extended to the case of narrowband signal and noise models. A useful class of non-Gaussian narrowband noise models is developed for which the locally optimum receiver implementation is explicitly determined. These structures are shown to provide considerable improvement over conventional linear receiver structures. The basis of comparison here is the asymptotic relative efficiency (A.R.E.). Some fairly simple suboptimum receiver structures are demonstrated to provide fairly efficient and robust performance in the presence of non-Gaussian narrowband noise.

INTRODUCTION

The concept of a locally optimum receiver, subject to appropriate qualifications, provides a rather general approach to nonlinear receiver design in the presence of non-Gaussian noise. Here the approach is to determine fairly general receiver structures appropriate for a wide range of non-Gaussian noise models under the assumption of weak input signal levels. The argument is that by optimizing performance in the critical weak signal regime, adequate performance will be provided in the less critical moderate-to-large signal level regime. At present there is little quantitative information to refute this argument.

Previous work (cf. [1]-[3]) has concentrated upon lowpass signal and noise models. In many situations, however, narrowband signal and noise models are more appropriate. This paper then is concerned with the extension of the concept of a locally optimum receiver structure to cover this case. More specifically, we will be concerned with the implementation and performance of locally optimum receiver structures for the coherent detection of narrowband signals additively embedded in non-Gaussian noise. Careful consideration will be given to the problem of modeling non-Gaussian narrowband noise processes together with the explicit determination of the corresponding locally optimum receiver structures and evaluation of the resulting performance. The performance criterion in all cases will be the asymptotic relative efficiency (A.R.E.). This is an asymptotic comparison of the relative sample sizes required by two detectors to achieve identical false alarm probability α and detection probability β in the limit of large sample sizes and vanishingly small input signals.

LOCALLY OPTIMUM RECEIVER STRUCTURE

Consider the single channel test of a null hypothesis (H_0) of no signal or noise alone present against an alternative hypothesis (H_1) of a known signal in addition to noise present. The observables will be assumed to consist of the sequence of complex amplitudes[††] $\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_N$ whose real

---

[†] This work was supported in part by the Office of Naval Research under Contract No. N00014-75-C-0281.

[††] Complex quantities will be consistently identified by a tilde over the symbol. In particular $\tilde{r}_i = r_{ci} - j r_{si}$ $i=1,2,\ldots,N$ with r_{ci}, r_{si} the inphase and quadrature components respectively.

and imaginary parts are obtained by sampling respectively the outputs of appropriate inphase and quadrature channel filters. We have then the hypothesis test

$$H_0 : \underline{\tilde{r}} = \underline{\tilde{n}} \qquad \text{vs.} \qquad H_1 : \underline{\tilde{r}} = A\,\underline{\tilde{S}}+\underline{\tilde{n}} \tag{1}$$

where $\underline{\tilde{n}}^T = (\tilde{n}_1, \tilde{n}_2, \ldots, \tilde{n}_N)$ is a complex noise vector of dimension N representing the inphase and quadrature samples of a narrowband noise process n(t). Similarly, $\underline{\tilde{S}}^T = (\tilde{S}_1, \tilde{S}_2, \ldots, \tilde{S}_N)$ is a complex N-vector representing a sequence of N samples of the assumed known complex signal amplitude. The quantity A appearing in (1) is an unknown signal amplitude assuming positive real values. It is to be noted that this model implicitly assumes coherent reception.

The optimum detection strategy is assumed based upon a threshold test of the likelihood ratio, i.e.,

$$\Lambda_N(\underline{\tilde{R}};A) = \frac{p_{\underline{\tilde{r}}|H_1}(\underline{\tilde{R}}|H_1)}{p_{\underline{\tilde{r}}|H_0}(\underline{\tilde{R}}|H_0)} \gtrless \Lambda_0 \tag{2}$$

where Λ_0 is some appropriately defined decision threshold and need not be the same from one line to the next. We will assume that the noise samples are independent and identically distributed (i.i.d.) so that

$$p_{\underline{\tilde{r}}|H_0}(\underline{\tilde{R}}|H_0) = \prod_{i=1}^{N} p_{n_c,n_s}(R_{ci},R_{si}) \tag{3}$$

where $p_{n_c,n_s}(\cdot,\cdot)$ is the joint probability density function (p.d.f.) of the inphase and quadrature noise components and R_{ci}, R_{si} represent actual observations on the output of the inphase and quadrature channels respectively. Similarly, under H_1 we have

$$p_{\underline{\tilde{r}}|H_1}(\underline{\tilde{R}}|H_1) = \prod_{i=1}^{N} p_{n_c,n_s}(R_{ci}-AS_{ci},R_{si}-AS_{si}) \tag{4}$$

so that the likelihood ratio becomes

$$\Lambda_N(\underline{\tilde{R}};A) = \prod_{i=1}^{N} \frac{p_{n_c,n_s}(R_{ci}-AS_{ci},R_{si}-AS_{si})}{p_{n_c,n_s}(R_{ci},R_{si})} \tag{5}$$

The receiver is readily implemented as a threshold test on the log-likelihood ratio provided the joint p.d.f. $p_{n_c,n_s}(\cdot,\cdot)$ is specified and additionally both A and $\underline{\tilde{S}}$ are known at the receiver. Often the true value of A is small but otherwise unknown. In such cases it makes sense to employ the locally most powerful or locally optimum receiver structure. This is precisely the detector structure which maximizes the derivative of the power function at A=0 subject to a fixed false alarm probability α. From the extended Neyman-Pearson Lemma (cf. [4]) it can be shown that this is the same as the threshold test

$$\left.\frac{\partial\Lambda_N(\underline{\tilde{R}};A)}{\partial A}\right|_{A=0} \gtrless \Lambda_0 \tag{6}$$

Observe that

$$\left.\frac{\partial\Lambda_N(\underline{\tilde{R}};A)}{\partial A}\right|_{A=0} = -\sum_{i=1}^{N} \frac{\left\{S_{ci}\dfrac{\partial p_{n_c,n_s}(R_{ci},R_{si})}{\partial R_{ci}} + S_{si}\dfrac{\partial p_{n_c,n_s}(R_{ci},R_{si})}{\partial R_{si}}\right\}}{p_{n_c,n_s}(R_{ci},R_{si})} \tag{7}$$

For example, suppose that the inphase and quadrature noise components are statistically independent, i.e.,

$$p_{n_c,n_s}(R_{ci},R_{si})=p_{n_c}(R_{ci})p_{n_s}(R_{si}) \tag{8}$$

The resulting decision statistic then becomes

$$T_N(\underline{\tilde{R}}) = \sum_{i=1}^{N} \{S_{ci}g_c(R_{ci})+S_{si}g_s(R_{si})\} \tag{9}$$

where

$$g_c(R) = -\frac{d}{dR}\ln p_{n_c}(R) \tag{10a}$$

and

$$g_s(R) = -\frac{d}{dR}\ln p_{n_s}(R) \tag{10b}$$

Here it is clear that the locally optimum receiver passes the inphase and quadrature components separately through zero-memory nonlinear (ZMNL) characteristics represented by $g_c(\cdot)$ and $g_s(\cdot)$ respectively followed by matched filtering. In the absence of a quadrature signal component (i.e., $S_{si}=0$ $i=1,2,\ldots,N$) the receiver structure represented by (3) reduces to the familiar locally optimum detector obtained on the basis of lowpass representations (cf. [1]-[3])

In the more general case, the inphase and quadrature noise components are not statistically independent and the locally optimum detector structure requires joint nonlinear processing of both components. In what follows we will assume that the joint p.d.f. of the inphase and quadrature noise components possesses circular symmetry such that

$$p_{n_c,n_s}(R_c,R_s) = f\left(\sqrt{R_c^2+R_s^2}\right) \tag{11}$$

i.e., the joint p.d.f. depends only upon the amplitude or envelope $R=\sqrt{R_c^2+R_s^2}$. It follows that

$$\frac{\partial p_{n_c n_s}(R_c,R_s)}{\partial R_c} = \frac{R_c}{R}\;\frac{df(R)}{dR} \tag{12a}$$

and similarly

$$\frac{\partial p_{n_c,n_s}(R_c,R_s)}{\partial R_s} = \frac{R_s}{R}\;\frac{df(R)}{dR} \tag{12b}$$

The test statistic corresponding to the locally optimum threshold test defined by (7) then becomes

$$T_N(\underline{\tilde{R}}) = \sum_{i=1}^{N} \mathrm{Re}\{\tilde{R}_i\tilde{S}_i^*\}g(R_i) \tag{13}$$

where

$$g(R) = -\frac{d\ln f(R)}{RdR} \triangleq g_{\ell o}(R) \tag{14}$$

and† $R_i \triangleq \sqrt{R_{ci}^2+R_{si}^2}$ $i=1,2,\ldots,N$ is the envelope of the i'th observation with complex amplitude $\tilde{R}_i=R_{ci}-jR_{si}$. The resulting receiver structure is illustrated in Fig. 1. Note that even in the absence of a quadrature signal component, the locally optimum receiver structure in general depends on the quadrature channel observations through the envelope. In the particular case of Gaussian noise it is easily established that $g_{\ell o}(R) = 1/\sigma^2$ where σ^2 is the common variance of the inphase and quadrature components. It follows that in the absence of a quadrature signal component the quadrature channel observations can be neglected. Furthermore, if we restrict attention to circularly symmetric distributions as described by (11), it can be shown[5] that the inphase and quadrature components are statistically independent only in the Gaussian case. As a result, for non-Gaussian narrowband noise, the quadrature observation cannot be neglected and the resulting locally optimum receiver structure differs from that obtained assuming lowpass signal and noise representations.

---

† It should cause no confusion that we are using R_i to represent an envelope value and $\tilde{R}_i$ to represent a complex amplitude.

CIRCULARLY SYMMETRIC NOISE DISTRIBUTIONS

We will consider two possible approaches to the development of two-dimensional circularly symmetric distributions as appropriate models for the joint distribution of inphase and quadrature components of non-Gaussian narrowband noise processes. The first and most obvious approach is to choose f(R) in (11) as functionally equivalent to some useful symmetric non-Gaussian univariate p.d.f.'s subject to the normalization constraint

$$\int_0^\infty R f(R)\,dR = \frac{1}{2\pi} \tag{15}$$

For example, consider the generalized Gaussian distribution with

$$f(R) = \frac{\alpha\Gamma\,(4/\alpha)}{4\pi\sigma^2\Gamma^2\,(2/\alpha)} \exp\{-\,(\frac{R}{c\sigma})^{\alpha}\,\};\ 0\leq R<\infty \tag{16}$$

where

$$c = \sqrt{\frac{2\Gamma(2/\alpha)}{\Gamma(4/\alpha)}} \tag{17}$$

and σ^2 is the common variance of the inphase and quadrature noise components. For $\alpha=2$ the inphase and quadrature noise components are clearly independently Gaussian distributed with zero means and common variance σ^2. The corresponding optimum ZMNL receiver characteristic from (14) is determined as

$$\sigma^2 g_{\ell o}(R) = \alpha\left[\frac{\Gamma\,(4/\alpha)}{2\Gamma(2/\alpha)}\right]^{\alpha/2} (\frac{R}{\sigma})^{\alpha-2} \tag{18}$$

which is illustrated in Fig. 2 as a function of (R/σ) for several values of α. Notice that $g_{\ell o}(R)$ is constant for $\alpha=2$ as expected while for $\alpha=1$, $g_{\ell o}(R)$ is proportional to $1/R$. The latter characteristic possesses an alternative implementation to that illustrated in Fig. 1. In particular, the locally optimum receiver could then be implemented as an ideal narrowband limiter followed by separate inphase and quadrature channel filtering.

An alternative approach is to assign the marginal distributions a specified symmetric distribution with common univariate characteristic function $\Phi(u) = \Psi(\sqrt{u^2})$ where $\Psi(\cdot)$ is some appropriately defined function. A circularly symmetric joint distribution can then be assigned to the inphase and quadrature noise components by defining the joint ch.f.

$$\Phi_{n_c,n_s}(u_1,\,u_2) = \Psi(\sqrt{u_1^2+u_2^2}\,) \tag{19}$$

It follows by inverse Fourier transformation (cf. [5] for details) that the corresponding joint p.d.f. is given by

$$p_{n_c,n_s}(R_c,R_s) = \frac{1}{2\pi}\int_0^\infty \Psi(\lambda)\lambda J_0(R\lambda)\,d\lambda = f(R) \tag{20}$$

where $R = \sqrt{R_c^2 + R_s^2}$ and $J_0(\cdot)$ represents the ordinary Bessel function of order zero. The function f(R) can then be obtained by performing the indicated integration.

As an example of this approach, suppose the inphase and quadrature noise components are chosen to be identically distributed zero-mean Cauchy variates with scale parameter β, i.e.,

$$p_{n_c}(x) = p_{n_s}(x) = \frac{1}{\pi\beta}\frac{1}{1+(x/\beta)^2} \tag{21}$$

with corresponding common univariate ch.f.

$$\Phi(u) = e^{-\beta|u|} \tag{22}$$

so that, in particular, $\Psi(\lambda) = e^{-\beta\lambda}$. It follows from (20) that

$$f(R) = \frac{\beta}{2\pi}\frac{1}{(\beta^2+R^2)^{3/2}} \tag{23}$$

and the corresponding locally optimum ZMNL characteristic satisfies

$$\beta^2 g_{\ell o}(R) = \frac{3}{1+(R/\beta)^2} \tag{24}$$

Another choice of univariate ch.f. that proves useful is

$$\Phi_n(u) = \left[\frac{1}{1+|u/c|^\alpha}\right]^n ; \quad n=1,2,\ldots \tag{25}$$

with $c^\alpha = 2n/\sigma^2$ and $\alpha>0$. This choice has been described previously in [5] and has an interpretation as the ch.f. associated with a suitably normalized sum of i.i.d. random variables with common ch.f. $\Phi_1(u)$. Observe that

$$\lim_{n\to\infty} \Phi_n(u) = e^{-\frac{\sigma^2}{2}|u|^\alpha} \tag{26}$$

which is precisely the ch.f. of the unique symmetric stable distribution of order α(cf. [6]). Our particular interest will be restricted to the case $\alpha=2$. We recognize $\Phi_1(u)$ then as the ch.f. associated with the Laplace distribution possessing zero mean and variance σ^2. The Laplace distribution has often been used as a model for non-Gaussian impulsive noise phenomenon. For $n>1$ we have a sequence of distributions converging ultimately to the Gaussian distribution. This choice has the distinct advantage of providing a family of distributions parametrized by n and providing behavior intermediate between the Laplace and Gaussian distributions.

The corresponding function f(R) computed according to (20) is

$$f(R) = \frac{c^{n+1}}{2^n\pi\Gamma(n)} R^{n-1}K_{n-1}(cR) \quad ; \quad n=1,2,\ldots \tag{27}$$

where $K_\nu(\cdot)$ is the modified Bessel function of the second kind of order ν (cf.[7]) and $c = \sqrt{2n/\sigma^2}$. The corresponding optimum ZMNL receiver characteristic is determined according to

$$\sigma^2 g_{\ell o}(R) = \frac{2n}{cR}\frac{K_{n-2}(cR)}{K_{n-1}(cR)} \quad ; \quad n=1,2,\ldots, \tag{28}$$

which is shown plotted in Fig. 3 as a function of cR for several values of n. Note the eventual but relatively slow convergence to the limiting case $g_{\ell o}(R) = 1/\sigma^2$ known to be correct for Gaussian noise.

ASYMPTOTIC RELATIVE EFFICIENCY

Under appropriate regularity conditions (cf. [8]), which will be assumed to be satisfied, the A.R.E. of a detector d_1 based upon the sequence of test statistics $\{T_N^{(1)}\}$ with respect to another detector d_2 employing the sequence of test statistics $\{T_N^{(2)}\}$ can be expressed as the ratio of their respective efficacies, i.e.,

$$\text{A.R.E.}_{1,2} = \frac{\mathcal{E}_1}{\mathcal{E}_2} \tag{29}$$

where the efficacy $\mathcal{E}_i$ is defined according to

$$\mathcal{E}_i = \lim_{N\to\infty} \frac{\left[\left. \frac{\partial^m E_A\{T_N^{(i)}\}}{\partial A^m} \right|_{A=0} \right]^2}{N \operatorname{var}_{H_0}\{T_N^{(i)}\}} \quad ; \quad i=1,2,.. \tag{30}$$

Here $E_A\{\cdot\}$ represents conditional expectation conditioned upon the signal amplitude A and m is the order of the first non-zero derivative of $E_A\{T_N^{(i)}\}$ i=1,2, evaluated at A=0.

If the sequence of test statistics $\{T_N(\tilde{R})\}$ is defined according to (13) for arbitrary choice of ZMNL receiver characteristic $g(\cdot)$ it is easily shown (cf. [9] for details) that

$$\left. \frac{\partial E_A\{T_N(\underline{\tilde{R}})\}}{\partial A} \right|_{A=0} = -\pi E_s \int_0^\infty R^2 g(R) f'(R)\,dR \tag{31}$$

where

$$E_s = \sum_{i=1}^{N} \operatorname{Re}\{\tilde{S}_i \tilde{S}_i^*\} = \sum_{i=1}^{N} [S_{ci}^2 + S_{si}^2] \tag{32}$$

provides a measure of signal energy. Similarly we have

$$\operatorname{var}_{H_0}\{T_N(\underline{\tilde{R}})\} = \sum_{i=1}^{N} E_{H_0}\{\operatorname{Re}^2\{\tilde{S}_i^* \tilde{R}_i\}\, g^2(R_i)\} = \pi E_s \int_0^\infty R^3 g^2(R) f(R)\,dR \tag{33}$$

so that according to (30) the corresponding efficacy is then given by

$$\mathcal{E} = \lim_{N\to\infty} \frac{E_s [\int_0^\infty R^2 g(R) f'(R)\,dR]^2}{N \int_0^\infty R^3 g^2(R) f(R)\,dR} \tag{34}$$

It follows after some algebra, the details of which are provided in [9], that the A.R.E. of the locally optimum receiver structure with respect to the linear receiver is given by

$$\text{A.R.E.}_{\ell o,\ell} = \pi\sigma^2 \int_0^\infty R^3\, g_{\ell o}^2(R) f(R)\,dR \tag{35}$$

For example, if the noise possesses generalized Gaussian distribution as defined by (16) and the sequel we obtain

$$\text{A.R.E.}_{\ell o,\ell} = \frac{\sigma^2 \Gamma(4/\alpha)}{4\Gamma^2(2/\alpha)} \tag{36}$$

which is plotted as a function of α in Fig. 4. Note the relatively broad minimum around $\alpha=2$ where the linear receiver is optimum and the A.R.E. is unity as expected. Also illustrated in Fig. 4 is the corresponding A.R.E. of the hard-limiter or clipper (i.e., $g(R) = 1/R$) compared to the linear receiver which is computed according to

$$\text{A.R.E.}_{c,\ell} = \frac{\Gamma(4/\alpha)\Gamma^2(1/\alpha)}{4\Gamma^3(2/\alpha)} \tag{37}$$

Notice that the A.R.E.'s are equal as expected for $\alpha=1$ while for $\alpha=2$ corresponding to the Gaussian case the A.R.E. provided by the clipper is $\pi/4 \simeq .785$. The clipper is relatively efficient over a broad range of α. Similarly, for generalized Laplace noise it is easily established that

$$\text{A.R.E.}_{c,\ell} = \frac{\pi}{4} \left[\frac{\sqrt{n}\, \Gamma(n-1/2)}{\Gamma(n)} \right]^2 \tag{38}$$

For example, for n=1 corresponding to Laplace noise we have a value $\pi^2/4 \approx 2.47$. The limiting value for large n is easily established as $\pi/4$ with the aid of the known result (cf.[7], pg. 257)

$$\lim_{n\to\infty} \sqrt{n}\,\frac{\Gamma(n-1/2)}{\Gamma(n)} = 1 \tag{39}$$

The resulting A.R.E. for several intermediate values of n is provided in Table 1

n	$\text{A.R.E.}_{c,\ell}$
1	2.467
2	1.234
3	1.040
4	0.964
5	0.922
6	0.897
10	0.849
∞	0.785

Table 1

A.R.E. of Clipper Relative to Linear Receiver for Generalized Laplace Noise.

Finally, as an example of the relative efficiency of this structure for circularly symmetric Cauchy noise the A.R.E. of the clipper with respect to the locally optimum receiver is easily shown to be

$$\text{A.R.E.}_{c,\ell o} = \frac{15}{18} \approx 0.833 \tag{40}$$

indicating little loss in asymptotic efficiency.

SUMMARY AND CONCLUSIONS

The structure of locally optimum receivers for narrowband signals in non-Gaussian noise has been examined in some detail. A useful class of non-Gaussian narrowband noise processes has been introduced whose inphase and quadrature components possess circularly symmetric joint distribution and for which the locally optimum receiver structure has been explicitly determined. Results are also provided allowing evaluation of the A.R.E. of arbitrary nonlinear receiver structures for the detection of known signals embedded in narrowband noise of this class. While the locally optimum receiver provides significant improvements in A.R.E. compared to conventional linear processors, it is often difficult to implement. A particularly simple suboptimum receiver structure shown to be equivalent to a narrowband limiter, on the other hand, is shown to provide relatively efficient and robust performance for a wide range of non-Gaussian noise assumptions.

REFERENCES

1. Capon, J., "On the Asymptotic Relative Efficiency of Locally Optimum Detectors", IRE Trans. Inform. Theory, Vol. IT-7, No. 2, April 1961, pp 67-72.
2. Capon, J., "Locally Optimum Nonparametric Methods for the Detection of Signals in Noise", Proceedings of the 1963 IEEE Symposium on Adaptive Processes, Sept. 1973, pp 64-77.

3. Miller, J.H., and Thomas, J.B., "Detectors for Discrete-Time Signals in Non-Gaussian Noise", IEEE Trans. Inform. Theory, Vol. IT-18, No. 2, March 1972, pp 241-250.

4. Lehmann, F. L., Testing Statistical Hypotheses, John Wiley and Sons, Inc. New York, 1959.

5. Modestino, J.W., "Nonparametric and Adaptive Detection of Dependent Data" Ph.D. Thesis, Electrical Engineering Dept., Princeton University, 1969.

6. Feller, W., An Introduction to Probability Theory and Its Applications, Vol. II, John Wiley & Sons, Inc., New York, 1965.

7. Abramowitz, M., and Stegun, I.A., Handbook of Mathematical Functions, National Bureau of Standards, AMS 55, 1964.

8. Fraser, D.A.S., Nonparametric Methods in Statistics, John Wiley and Sons New York, 1951.

9. Modestino, J.W., "Locally Optimum Receiver Structures for Known Signals in Non-Gaussian Narrowband Noise", unpublished R.P.I. report.

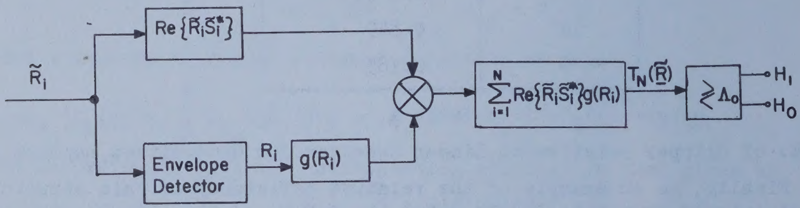

$\tilde{R}_i$
$\mathrm{Re}\{\tilde{R}_i\tilde{S}_i^*\}$
Envelope Detector
R_i
$g(R_i)$
$\sum_{i=1}^{N}\mathrm{Re}\{\tilde{R}_i\tilde{S}_i^*\}g(R_i)$
$T_N(\tilde{\underline{R}})$
$\gtrless \Lambda_o$
H_1
H_0

Figure 1
Locally Optimum Receiver Structure when Inphase and Quadrature Noise Components Possess Circularly Symmetric Joint Distribution

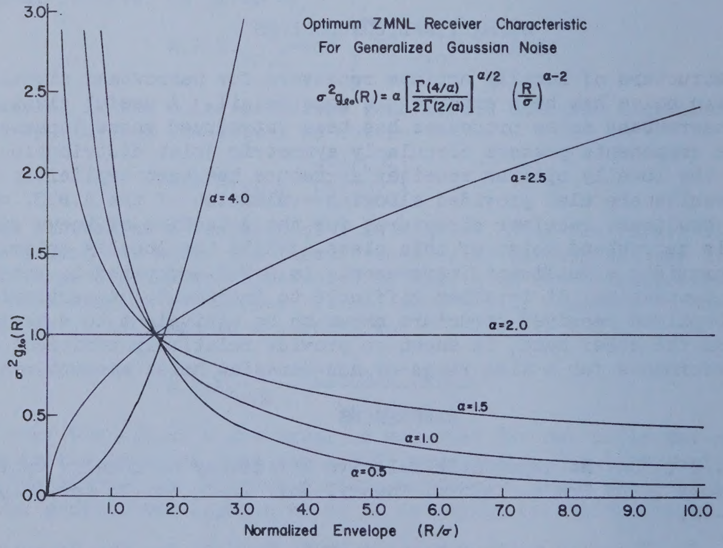


Figure 2
Optimum ZMNL Receiver Characteristic for Generalized Gaussian Noise

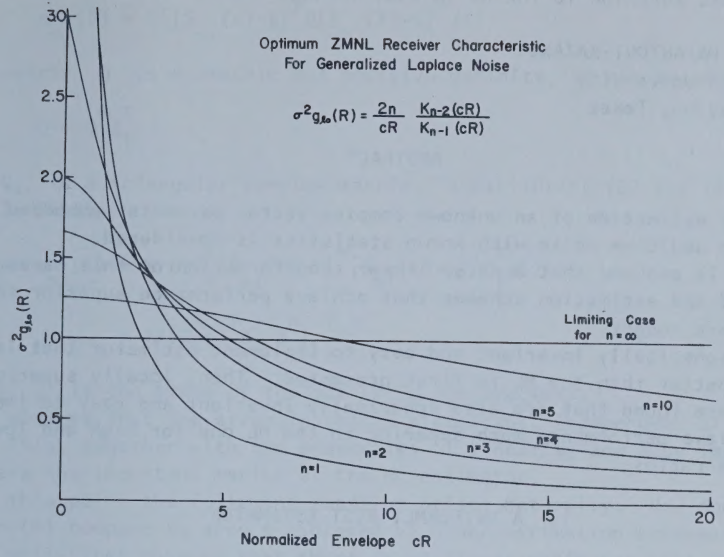


Figure 3

Optimum ZMNL Receiver Characteristic for Generalized Laplace Noise

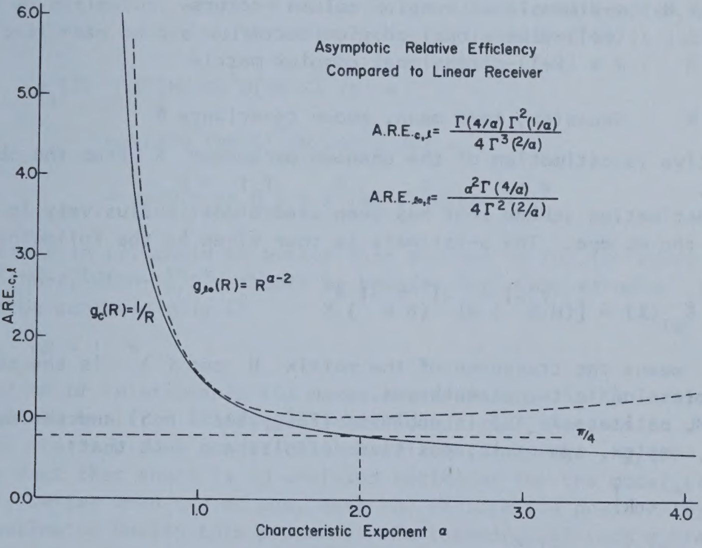


Figure 4

Asymptotic Relative Efficiency of Clipper Compared to Linear Receiver in Presence of Generalized Gaussian Noise

ESTIMATORS SUPERIOR TO THE ML IN MULTIVARIATE NORMAL MODELS*

P. PAPANTONI-KAZAKOS
Rice University
Houston, Texas

ABSTRACT

The estimation of an unknown complex vector parameter embedded in Gaussian additive noise with known statistics is considered.

It is assumed that a noisy linear transformation of this parameter is observed and estimation schemes that achieve performance superior to the ML one are sought.

A spherically invariant and easy to implement estimator that is uniformly better than the ML is first presented. Then, locally superior estimators are found that are also spherically invariant and easy to implement and achieve performance much superior to the ML one for high and low signal-to-noise ratios.

1. A UNIFORMLY BEST ESTIMATOR

Let us consider the linear communication model given by expression

$$X = HS + N \tag{1}$$

where

X, N : m-dimensional complex column vectors
S : (k+1)-dimensional complex vector
H : m x (k+1)-dimensional complex matrix

and

N : Gaussian, zero mean, known covariance R

The objective is estimation of the unknown parameter S from the observations X.

The estimation scheme that has been used almost exclusively in such models is the ML one. The S-estimate is then given by the following expression:

$$\hat{S}_{ml}(X) = [(H^T R^{-1})^* H]^{-1} (H^T R^{-1})^* X \tag{2}$$

where H^T means the transpose of the matrix H and $(\ )^*$ is the conjugate of the expression in the parentheses.

The ML estimate in (2) is unbiased ($E\{\hat{S}_{ml}(X)/S\} = S$) and the matrices R, R^{-1} are complex, symmetric, positive definite and such that

$$R^* = R^T$$
$$(R^{-1})^T = (R^{-1})^* \tag{3}$$

The mean square conditional error $e_{ml}(S)$ of the maximum likelihood estimate in (2) is, in general, a function of a $(k+1)\times(k+1)$ complex weighting matrix Q. That is

---

*This work was supported by National Science Foundation Grant #ENG75-10516.

$$e_{ml}(S) = E\{[\hat{S}_{ml}(X)-S]^T Q[\hat{S}_{ml}(X)-S]^*/S\} \tag{4}$$

If the matrix Q is symmetric and positive definite, it can be expressed as

$$Q = Q_1^* Q_1^T \tag{5}$$

where Q_1 is a triangular complex matrix. Substituting (5) and (2) in (4), we easily obtain

$$\begin{aligned} e_{ml}(S) &= \mathrm{tr}E\{Q_1^*[\hat{S}_{ml}(X)-S]^*[\hat{S}_{ml}(X)-S]^T Q_1^T/S\} \\ &= \mathrm{tr}\{Q_1^*[H^T R^{-1} H^*]^{-1} Q_1^T\} \end{aligned} \tag{6}$$

where t{ } is the trace of the matrix in the brackets.

We observe that the error in (6) is a constant, independent of the S value. This, together with the properties of linearity and easy implementation are the important merits of the ML estimator.

At this point the following question arises naturally. How does the error in (6) compare to errors achieved by other estimation schemes?

We will first observe that there is no linear estimator that performs uniformly better than the ML one. Indeed if

$$\hat{S}_1(X) = MX$$

is such an estimator, where M a $(k+1) \times m$-dimensional complex matrix, the conditional mean square weighted error $e_1(S)$ achieved by it is

$$\begin{aligned} e_1(S) &= E\{[MX-S]^T Q[MX-S]^*/S\} = \\ &= \mathrm{tr}E\{Q_1^*[MX-S]^*[MX-S]^T Q_1^T/S\} = \\ &= \mathrm{tr}\{Q_1^* M^* R M^T Q_1^T\} + S^T[MH-I]^T Q[MH-I]^* S^* \end{aligned} \tag{7}$$

If the error in (7) could be better than the one in (6) for every S, the term $S^T[MH-I]^T Q[MH-I]^* S^*$ should be bounded for every value of S. But that can be achieved only if

$$MH = I \tag{8}$$

Satisfaction of relationship (8) makes the estimator $\hat{S}_1(X)$ unbiased and it is well known (from Cramér-Rao lower bound) that there is no unbiased estimator that achieves conditional error lower than the $e_{ml}(s)$.

The fact that there is no unbiased estimator for the model (1) that is uniformly better than the ML one, does not excluse the possibility of a biased estimator having this property. Furthermore, if such a biased estimator were also Bayes for some a priori S statistics, its mean square conditional performance would be impossible to improve upon for every value of S (uniformly). That is, this estimator would then be admissible.

If S is chosen Gaussian, zero mean with autocovariance matrix $D(\lambda)$, where λ scalar with distribution $\varphi(\lambda) = (1-\lambda)/\lambda$ for $\lambda \in [0,1]$, then an admissible estimator that is also uniformly better than the ML is obtained. The result is expressed by the following theorem.

Theorem 1

For the general model in (1), let E_m be the maximum eigenvalue of the complex positive definite and symmetric matrix $H^T R^{-1} H^*$. Then, if the matrix $H^T R^{-1} H^*$ is such that

$$\frac{\mathrm{tr}\{[H^T R^{-1} H^*]^{-1}\}}{E_m} > \frac{k+1}{2} + 2 \tag{T1}$$

and if we can find a constant b such that

$$0 < b < 1$$

$$\frac{\mathrm{tr}\{[H^T R^{-1} H^*]^{-1}\}}{E_m} \geq \frac{k+1}{2} + 3 - b \tag{T2}$$

there is an admissible nonlinear estimator $\hat{S}(X,b)$ that is uniformly better than the ML one and is given by the following expression

$$\hat{S}(X,b) = \left[1 - \left(\frac{k+3-2b}{h(X)} - \frac{2\exp\{-\frac{1}{2}h(X)\}}{h(X)\int_0^1 \lambda^{(k+1)/2-b}\exp\{-\frac{\lambda}{2}h(X)\}d\lambda}\right)\right] \cdot$$

$$\cdot\ [H^{*T} R^{-1T} H]^{-1} H^{*T} R^{-1T} X \tag{T3}$$

where

$$h(X) = X^T R^{-1} H^* [H^T R^{-1} H^*]^{-1} H^T R^{-1} X^* \tag{T4}$$

The estimator $\hat{S}(X,b)$ is such that

$$E\{[\hat{S}(X,b)-S]^T [\hat{S}(X,b),S]^* / S\} \leq \mathrm{tr}\{[H^T R^{-1} H^*]^{-1}\} \quad \forall\, S$$

and it is spherically invariant. Also, it is obviously easy to implement. For certain R,H matrices the constant b can be chosen such that $(k+1)/2 - b$ is an integer. In this case, the integral in expression (T3) can be found analytically.

Condition (T1) is indicative of the systems in (1) for which an $\hat{S}(X,b)$ estimator can be found. Since the trace of any matrix is bounded from below by its smaller eigenvalue times the dimensionality, we have

$$\Sigma \text{ eigenvalues} = \mathrm{tr}\{[H^T R^{-1} H^*]^{-1}\} \geq (k+1)\frac{1}{E_m} \tag{9}$$

where $(k+1)$ is the dimensionality of the matrix $H^T R^{-1} H^*$ and E_m its largest eigenvalue. Since $H^T R^{-1} H^*$ is symmetric and positive definite all its eigenvalues are real.

Because of (9) we conclude that a sufficient condition for (T1) and (T2) to be satisfied for some $b: 0 < b < 1$, is that

$$\left[\frac{2}{E_m^2} - 1\right](k+1) \geq \max(4, 4+2(1-b)) = 4 + 2(1-b) \tag{10}$$

for some $b: 0 < b < 1$

We should point out here that the dimensionality (k+1) of the parameter S does not have to satisfy condition (9) to satisfy (T1) and (T2). Matrices H and R may exist with dimensionality k+1 lower than 5 that still satisfy these two conditions. It turns out that the estimator $\hat{S}(X,b)$ in (T3) is impossible to improve upon uniformly, where the criterion of evaluation is the weighted mean square error with arbitrary Hermitian weighting matrix. This result is formally expressed by the following lemma.

Lemma

For $\hat{S}(X)$ any S-estimator and Q any positive definite and symmetric (Hermitian) complex matrix, the estimator $\hat{S}(X,b)$ in (T4) satisfies the inequality

$$E\{[\hat{S}(X,b)-S]^T Q[\hat{S}(X,b)-S]^*/S\} \leq E\{[\hat{S}(X)-S]^T Q[\hat{S}(X)-S]^*/S\}$$

$$\forall\ S$$

With the above lemma we conclude our search for an estimator that achieves a uniformly good performance for the model in (1). The estimator $\hat{S}(X,b)$ in (T3) achieves mean square performance that cannot be exceeded by any other estimator.

2. LOCALLY SUPERIOR ESTIMATORS

In the previous section, a uniformly superior estimator was presented for the linear, Gaussian model in (6). In this section we will present an estimator that depends on two arbitrary parameters. Proper choice of these parameters results in superior performance of the estimator inside different signal-to-noise ranges.

Observing the estimator in (T3) one has the natural tendency of taking a simpler version of it. The uniformly superior properties will stop existing then, but new local and much more superior qualities might appear. The simplification that one might first attempt is substitution of the parentheses in (T3) by an exponential expression only, introducing at the same time two parameters a and b .

The estimator obtained this way is

$$\hat{S}(X) = [1 - a\exp\{-\frac{b}{2}h(X)\}][H^{*^T}R^{-1^T}H]^{-1}H^{*^T}R^{-1^T}X] \tag{11}$$

where h(X) is given by (T4).

It is found that the bias of the estimator in (11) and the ratio of its mean square weighted error over the corresponding ML one are given by

$$B(\hat{S}(X)) = \frac{|a|}{(1+b)^{3/2}}\exp\{-\frac{b}{2(1+b)}S^T W_1^T W_1^* S^*\}S = b(S)\cdot S \tag{12}$$

$$\frac{e(S)}{C} = a^2\Big[\Big(1 - \frac{S^T Q S^*}{C(1+2b)}\Big)\frac{1}{(1+2b)^2}\exp\{-\frac{1}{1+2b}S^T R_1^{-1} S^*\}$$

$$- \frac{S^T Q S^*}{C(1+b)^3}\exp\{-\frac{b}{1+b}S^T R_1^{-1} S^*\}\Big]$$

$$- a\,\frac{2}{(1+b)^{3/2}}\exp\{-\frac{b}{2(1+b)}S^T R_1^{-1} S^*\} + 1 \qquad (13)$$

where $Q = Q_1^T Q_1^*$ the weighting matrix, $R_1 = [H^T R^{-1} H^*]^{-1}$, $R_1^{-1} = W_1^T W_1^*$, $C = \mathrm{tr}\{Q_1^*[H^T R^{-1} H^*]^{-1} \cdot Q_1^T\}$ and b must be larger than $-\frac{1}{2}$. The study for the proper choice of the parameters a and b in (11) will be based on the measures $b(S)$ and $[e(S)]/C$ in (12) and (13) correspondingly. An estimator $\hat{S}(X)$ that performs well inside a certain signal-to-noise range, achieves low $[e(S)]/C$ at the expense of an also low $b(S)$ in it.

i) High Signal-to-Noise Ratio. By constructive analysis it was found that both the bias $b(S)$ and the ratio $[e(S)]/C$ can be kept very low for K large enough if the parameters a and b are properly chosen. K is the low bound for $(S^T Q S^*)/C$.

Specifically, one may choose an arbitrary upper bound r_o for the ratio $[e(S)]/C$, as small as desired. Then, if K is large enough to satisfy

$$K(1-r_o) \gg 1$$

the upper bound q for the bias $b(S)$ can be chosen equal to

$$q = \sqrt{\frac{1-r_o}{K}} \qquad (14)$$

which is a very small number for large K.

The parameter b is then chosen

$$b = \frac{K-1}{2} \qquad (15)$$

and one choice for the parameter a is

$$a = \min_{\substack{S \text{ in range}\\ \text{considered}}} q(1+b)^{3/2}\exp\{-\frac{b}{2(1+b)}S^T R_1^{-1} S^*\} =$$

$$= \min_{\substack{S \text{ in range}\\ \text{considered}}} \sqrt{\frac{(1-r_o)}{2^3}\,\frac{(K+1)^3}{K}}\;\exp\{-\frac{K-1}{2(K+1)}S^T H^T R^{-1} H^* S^*\} \qquad (16)$$

If the signal-to-noise ratio is also bounded from above so that

$$\frac{S^T H^T R^{-1} H^* S^*}{C} < L$$

where L a known large constant, then the proper choice for a is

$$a = \sqrt{\frac{1-r_o}{2^3} \cdot \frac{(K+1)^3}{K} \exp\{- \frac{K-1}{2(K+1)} CL\}} \qquad (17)$$

where

$$C = tr\{Q_1^* [H^T R^{-1} H^*]^{-1} Q_1^T\} \qquad (18)$$

ii) Low Signal-to-Noise Ratio. It is assumed that the signal-to-noise ratio keeps in levels low enough to satisfy

$$\frac{S^T H^T R^{-1} H^* S^*}{tr\{Q_1^* [H^T R^{-1} H^*]^{-1} Q_1^T\}} \leq K \qquad (19)$$

Then, the upper bound q on the bias can be chosen arbitrarily in which case the upper bound r_o on the ratio $[e(S)]/C$ should be such that

$$1 - q > r_o > 1 - 2q \qquad (20)$$

The parameter b is then chosen such that

$$\min(q^{-1},(1-r_o)^{-1}) \geq \frac{(1+b)^3}{(1+2b)^2} \exp\{K \cdot C \frac{b^2}{(1+b)(1+2b)}\} \qquad (21)$$

where C is given by (44) and (as pointed out in appendix B) b is larger than $-\frac{1}{2}$.

One choice that satisfies (47) is

$$b : \min(q^{-1},(1-r_o)^{-1}) \cdot \exp\{-KC\} \geq \frac{(1+b)^3}{(1+2b)^2} \qquad (22)$$

Finally, the parameter a can be chosen equal to

$$\min_{\substack{S \text{ in range} \\ \text{considered}}} q(1+b)^{3/2} \exp\{\frac{b}{2(1+b)} S^T R_1^{-1} S^*\}$$

which is equivalent to

$$a = q(1+b)^{3/2} \qquad (23)$$

if the S range starts from zero, where q is chosen and b is given by (22). If the S range is such that $(S^T H^T R^{-1} H^* S^*)/C$ is larger than a constant L , then a will be chosen equal to $q(1+b)^{3/2} \exp\{bC/[2(1+b)]L\}$ instead, where C is given by (18).

It is interesting to observe that in the low signal-to-noise ratio case, the bounds r_o and q cannot be kept simultaneously low. Indeed, because of relationship (20), if the bias is kept very low, the error is higher and vice versa. In any case, at the expense of a bias that can be very low, an error $e(S)$ that is lower (much lower sometimes) that C (the error achieved by the ML estimator) can be obtained. The estimator that realizes that is the estimator $\hat{S}(X)$ in (11), where the parameters b and a are given by (22) and (23) correspondingly.

REFERENCES

1. Goldstein, G.B. and P. Swerling, "Bootstrapping the generalized least-squares estimator in colored Gaussian noise with unknown covariance parameters," IEEE Trans. Inf. Th. IT-16: 385-392, July 1970.

2. Mohajeri, M., "A new estimator for an unknown signal imbedded in additive Gaussian noise," IEEE Trans. Inf. Th. IT-20: 181-190, March 1974.

3. Capon, J., R.J. Greenfield, and R.S. Kolker, "Multidimensional maximum-likelihood processing of large aperture seismic array," Proc. IEEE 55: 192-211, February 1967.

4. Mazo, J.E. and J. Salz, "Theory of error rates for digital FM," B.S.T.J. XLV: 1511-1535, 1966.

5. Bennett, W.R. and J. Salz, "Binary data transmission by FM over a real channel," B.S.T.J. 42: 2387-2426, 1963.

6. Papantoni-Kazakos, P. and I. Paz, "The performance of a digital FM system with discriminator: Intersymbol interference effects," IEEE Trans. in Communications COM-9(12), September 1975.

7. Hodges, J.L., Jr., and E.L. Lehman, "Some applications of the Cramér-Rao inequality," Proc. Second Berkeley Symp. Math. Stat. Prob.: 13-22, 1951.

8. Stein, C., "Inadmissibility of the usual estimator for the mean of a multivariate normal distribution," Proc. Third Berkeley Symp. Math. Stat. Prob. 1: 197-207, 1954.

9. Strawderman, W., "Proper Bayes minimax estimators of the multivariate normal mean," Ann. Math. Stat. 42: 385-388, 1971.

10. Strawderman, W., "On the existence of proper minimax estimators of the mean of a multivariate normal distribution," Proc. Sixth Berkeley Symp. Math. Stat. Prob. 1: 51-55, 1972.

11. Bock, M.E., "Minimax estimators of the mean of a multivariate normal distribution," Ann. Math. Stat. 3: 209-219, January 1975.

ON A RECEIVER FOR CHANNELS WITH MEMORY [†]

PRAMOD K. VARSHNEY and ABRAHAM H. HADDAD
Coordinated Science Laboratory and Department of Electrical Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

ABSTRACT

In this paper, the receiver design problem for channels with memory is considered. Optimal and suboptimal decision schemes are obtained for binary communication on a fading channel in the absence of intersymbol interference. On-off keying is considered as an example and a receiver with one-bit memory is shown to perform better than a memoryless receiver.

I. INTRODUCTION

In digital communication systems, receiver design is an important problem. Standard memoryless detection schemes assume the independence of the received data bits and a bit-by-bit detection is performed. However, in most practical digital communication systems, the independence assumption is not valid. The major sources of dependence in the received data sequence are fading, intersymbol interference and correlated noise. In systems where coding techniques are employed, transmitted bits are also dependent. In this paper, we consider only those communication systems where the dependence is introduced by the channel. The objective is to design a receiver with memory which would utilize the dependence in the received sequence to yield better performance.

Considerable attention has been given in the literature to the design of receivers with memory for communication in the presence of intersymbol interference. Chang and Hancock [1] devised an optimum decision procedure in which a decision was made on the basis of the complete message. Abend and Fritchman [2] modified the procedure and derived an optimum sequential compound detector under a fixed delay constraint. However, we shall not consider intersymbol interference as a source of channel memory and shall assume its absence. Several finite-memory algorithms as applied to estimation and detection problems have been reported in the literature [3-7]. In all of these algorithms, a finite memory is employed to generate sufficient statistics which, in turn, is used for the relevant estimation or detection problem. The adaptive detection scheme proposed by Baxa and Nolte [5] is similar to the problem considered in this paper. However, their approach is based only on the input-output characteristics of the digital channel, and not on the physical processes involved.

In this paper, a detection scheme with one-bit memory is proposed, whereby the decision on a particular bit is based also on the decision made on the previous bit. It is expected that introducing larger memory into the receiver will yield a better performance but at the expense of implementation complexity. The objective is, therefore, to design a system which is simple to implement and performs better than the zero-memory scheme. In Section two, the problem is defined and the appropriate assumptions are made. In the next two sections, optimal and suboptimal solutions to the problem are discussed. Finally, a special case is considered as an example in the last section.

[†] This work was supported by the Joint Services Electronics Program,(U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-72-C-0259.

II. PROBLEM STATEMENT

The problem considered is to design a receiver with one-bit memory for communication channels with memory. It is assumed that a binary system is operating over an amplitude fading channel in the absence of intersymbol interference. The bits being transmitted are assumed to be independent of each other with zeroes and ones equiprobable. The transmitted signal is $s_0(t)$ or $s_1(t)$ depending on whether a zero or one is to be sent. The noise is assumed to be white Gaussian with power spectrum N_0. Fading is assumed to be slow so that the channel gain is constant over the period of one bit and thus may be represented by a sequence of random variables $\{A_i\}$. The statistical dependence in the received sequence is introduced by the sequence $\{A_i\}$. This dependence as a function of fading is utilized to derive the detection scheme with memory. Let the received waveform during the kth bit be denoted by $r_k(t)$, then,

$$r_k(t) = A_k s_i(t) + n(t); \text{ if } H_i^k \text{ is true, } i = 0,1. \tag{1}$$

where H_0^k and H_1^k denote the hypothesis that a zero or a one is sent during the kth bit. A correlation receiver is employed to process the incoming waveform and the basic structure of the optimum receiver is as shown in Fig. 1. The outputs of the two integrators are denoted by $\{X_k\}$ and $\{Y_k\}$ respectively, i.e.,

$$X_k = \int_0^T r_k(t) s_0(t)dt \quad , \quad Y_k = \int_0^T r_k(t) s_1(t)dt. \tag{2}$$

The sequences $\{X_k\}$ and $\{Y_k\}$ are functions of the sequence $\{A_k\}$, and this relationship is utilized to design the receiver with memory.

III. AN OPTIMAL SOLUTION

In order to obtain the optimal solution for the one-bit memory detection problem, it is treated as a four-hypothesis Bayesian decision problem. The four hypotheses are defined as follows:

$$H_{00} : \; r_{k-1}(t) = A_{k-1} s_0(t) + n(t) \qquad r_k(t) = A_k s_0(t) + n(t) \tag{3a}$$

$$H_{01} : \; r_{k-1}(t) = A_{k-1} s_0(t) + n(t) \qquad r_k(t) = A_k s_1(t) + n(t) \tag{3b}$$

$$H_{10} : \; r_{k-1}(t) = A_{k-1} s_1(t) + n(t) \qquad r_k(t) = A_k s_0(t) + n(t) \tag{3c}$$

$$H_{11} : \; r_{k-1}(t) = A_{k-1} s_1(t) + n(t) \qquad r_k(t) = A_k s_1(t) + n(t). \tag{3d}$$

The sequences of random variables $\{X_k\}$ and $\{Y_k\}$ are as defined by (2). The costs are assigned based on the Hamming distance of the two received bits from the transmitted bits. The decision regions obtained by minimizing the Bayes' risk are described in the four-dimensional space defined by $\{X_k, X_{k-1}, Y_k, Y_{k-1}\}$. As a special case, on-off keying is considered.

Special Case (OOK):

For on-off keying, the four-dimensional space reduces to a two-dimensional space described by $\{Y_k, Y_{k-1}\}$, since $X_k = 0$, and the decision regions are easier to visualize. If the decision regions for the four hypotheses are denoted by Z_{ij}; i,j = 0,1, then the Bayes' risk may be expressed as

$$\Re = \iint_{Z_{00}} [f_{Y_{k-1}Y_k|H_{01}}(x,y|H_{01}) + f_{Y_{k-1}Y_k|H_{10}}(x,y|H_{10}) + 2f_{Y_{k-1}Y_k|H_{11}}(x,y|H_{11})]\,dxdy$$

$$+ \iint_{Z_{01}} [f_{Y_{k-1}Y_k|H_{00}}(x,y|H_{00}) + 2f_{Y_{k-1}Y_k|H_{10}}(x,y|H_{10}) + f_{Y_{k-1}Y_k|H_{11}}(x,y|H_{11})]\,dxdy$$

$$+ \iint_{Z_{10}} [f_{Y_{k-1} Y_k | H_{00}}(x,y|H_{00}) + 2f_{Y_{k-1} Y_k | H_{01}}(x,y|H_{01}) + f_{Y_{k-1} Y_k | H_{11}}(x,y|H_{11})]\, dxdy$$

$$+ \iint_{Z_{11}} [2f_{Y_{k-1} Y_k | H_{00}}(x,y|H_{00}) + f_{Y_{k-1} Y_k | H_{01}}(x,y|H_{01}) + f_{Y_{k-1} Y_k | H_{10}}(x,y|H_{10})]\, dxdy \quad (4)$$

where $f_{Y_{k-1} Y_k | H_{ij}}$ denotes the conditional density of Y_{k-1}, Y_k given H_{ij}. Note that the conditional densities of Y_k and Y_{k-1} given the gains (A_k, A_{k-1}) are Gaussian and thus their conditional density given H_{ij} may be derived if an appropriate model for the density of $\{A_k\}$ is assumed. If the likelihood ratios are defined as follows:

$$\Lambda_1 = f_{Y_{k-1} Y_k | H_{01}}(x,y|H_{01}) / f_{Y_{k-1} Y_k | H_{00}}(x,y|H_{00}) \quad (5a)$$

$$\Lambda_2 = f_{Y_{k-1} Y_k | H_{10}}(x,y|H_{10}) / f_{Y_{k-1} Y_k | H_{00}}(x,y|H_{00}) \quad (5b)$$

$$\Lambda_3 = f_{Y_{k-1} Y_k | H_{11}}(x,y|H_{11}) / f_{Y_{k-1} Y_k | H_{00}}(x,y|H_{00}) \quad (5c)$$

then the decision rule which minimizes the Bayes' risk is determined by the relations:

$$\Lambda_1 - \Lambda_2 + \Lambda_3 \underset{H_{01}, H_{10}}{\overset{H_{00}, H_{11}}{\lessgtr}} 1 \quad (6a)$$

$$\Lambda_2 - \Lambda_1 + \Lambda_3 \underset{H_{10}, H_{11}}{\overset{H_{00}, H_{01}}{\lessgtr}} 1 \quad (6b)$$

$$\Lambda_3 \underset{H_{11}}{\overset{H_{00}}{\lessgtr}} 1 \quad (6c)$$

$$\Lambda_1 \underset{H_{01}}{\overset{H_{10}}{\lessgtr}} \Lambda_2 \quad (6d)$$

Optimum decision regions Z_{ij} may be evaluated from the decision rule given in (6). It is observed that the optimum detection scheme is not easy to implement because the thresholds depend upon the level of the previous output. However, the decision regions are different from the ones for the memoryless scheme which indicates that an improvement is feasible if a detection scheme with memory is employed. Therefore, in the next section a suboptimal detection algorithm with memory is considered which keeps the essential features of the optimal scheme in that it minimizes the bit-error probability and is simpler to implement.

IV. SUBOPTIMAL SCHEME

The suboptimal scheme considered is a decision-feedback scheme for which it is assumed that the system error probability is sufficiently low so that the received digits can be assumed correct. Two different decision thresholds Q and R are employed depending upon the previous decision of the receiver as opposed to a single decision threshold T for the memoryless scheme. The structure of the proposed detector is shown in Fig. 2. If the output of the receiver is represented by the sequence $\{z_k\}$, the expression for the decision threshold during the kth bit can be written as follows:

$$T_k(z_{k-1}) = \begin{cases} Q & \text{if } z_{k-1} = 0 \\ R & \text{otherwise .} \end{cases} \tag{7}$$

The initial condition is

$$T_1 = T. \tag{8}$$

Threshold T is used for the detection of the first bit. Optimum values of Q and R could be computed at each step by the following necessary conditions:

$$\begin{aligned} \partial P(e_k = 1 | z_{k-1} = 0)/\partial Q = 0 \\ \partial P(e_k = 1 | z_{k-1} = 1)/\partial R = 0 \end{aligned} \tag{9}$$

where $\{e_k\}$ is the error sequence. The minimization of the conditional bit-error probability ensures the minimization of the bit-error probability. Eventually, a steady-state is reached and the steady-state values of the decision thresholds are employed for the practical implementation. Further details are furnished in the next section where an example is considered.

V. EXAMPLE

In this section, a binary on-off keying system is considered as an example. For this example, $s_0(t)$ is zero and $s_1(t)$ is a signal with energy E_s. The sequence of random variables $\{A_i\}$ is assumed to be a stationary Gaussian sequence with zero mean and variance σ_a^2. The correlation coefficient of the gains of two consecutive bits is denoted by r. The signal-to-noise ratio η and the correlation coefficient between the random variables Y_{k-1} and Y_k, ρ, are defined as follows:

$$\eta = (E_s^2 \sigma_a^2)(\sigma^2)^{-1} \equiv (E_s^2 \sigma_a^2)(N_0 E_s)^{-1} \tag{10}$$

$$\rho = (\eta r)(\eta + 1)^{-1} . \tag{11}$$

The conditional joint densities of Y_k and Y_{k-1} are Gaussian because of our assumptions. The decision thresholds for the optimal and suboptimal schemes are computed next.

Threshold Evaluation:

The optimum decision threshold for the memoryless scheme is given by

$$T = [(1+\eta)\ln(1+\eta)]^{\frac{1}{2}} \eta^{-\frac{1}{2}}. \tag{12}$$

The decision thresholds for the optimum scheme were shown to be different from T, and they were quite complex to implement. Therefore, the suboptimal scheme with memory is considered where threshold T is employed for the detection of the first bit. Thresholds Q_2 and R_2 for the detection of the second bit are computed from (9), for k = 2. The resulting set of equations which yield the values of Q_2 and R_2 are

$$\begin{aligned} &a/(a+b) + [\mathrm{erf}\{(T - \rho Q_2)(1+\eta)^{-\frac{1}{2}}(1-\rho^2)^{-\frac{1}{2}}\} \\ &+ \mathrm{erf}\{(T + \rho Q_2)(1+\eta)^{-\frac{1}{2}}(1-\rho^2)^{-\frac{1}{2}}\}]/2(a+b) = (1+\eta)^{\frac{1}{2}} \exp\{-\eta Q_2^2/2(1+\eta)\} \end{aligned} \tag{14}$$

$$\begin{aligned} &c/(c+d) + [\mathrm{erfc}\{(T - \rho R_2)(1+\eta)^{-\frac{1}{2}}(1-\rho^2)^{-\frac{1}{2}}\} \\ &+ \mathrm{erfc}\{(T + \rho R_2)(1+\eta)^{-\frac{1}{2}}(1-\rho^2)^{-\frac{1}{2}}\}]/2(c+d) = (1+\eta)^{\frac{1}{2}} \exp\{-\eta R_2^2/2(1+\eta\} \end{aligned} \tag{15}$$

where $\mathrm{erf}(x) = (2\pi)^{-\frac{1}{2}} \int_0^x \exp\{-t^2/2\} dt$

$$a = \mathrm{erf}(T) \qquad c = \mathrm{erfc}(T)$$

$$b = \mathrm{erf}(T(1+\eta)^{-\frac{1}{2}}) \qquad d = \mathrm{erfc}(T(1+\eta)^{-\frac{1}{2}}) .$$

Thresholds are computed in a recursive fashion and the equations yielding Q_k and R_k are similar to (14) and (15). The steady-state solution is obtained by setting $Q_{k+1} = Q_k = Q$ and $R_{k+1} = R_k = R$. The resulting equations cannot be solved analytically. However, approximate solutions for Q and R are obtained by linearization. Let ΔQ and ΔR represent the deviations of Q and R from the no-memory threshold T, i.e., $Q = T + \Delta Q$ and $R = T + \Delta R$. The approximate solution for the deviations ΔQ and ΔR is obtained from the linearized equations as a set of simultaneous linear equations:

$$a_{11}\Delta Q + a_{12}\Delta R = b_1 \quad , \quad a_{21}\Delta Q + a_{22}\Delta R = b_2 \tag{16}$$

where

$$\begin{aligned}
a_{11} &= u(a+b)(\alpha_2\beta + \alpha_3/\beta) - \alpha_1 (a+b)(2\pi)^{-\frac{1}{2}} + v\rho(c+d)(\alpha_3 - \alpha_2) + \\
&\quad + \eta T(a+b)(1+\eta)^{-1} \\
a_{12} &= v(c+d)(\alpha_2 + \alpha_3) - \alpha_1 (c+d)(2\pi)^{-\frac{1}{2}} \\
a_{21} &= \alpha_1 (a+b)(2\pi)^{-\frac{1}{2}} - v(a+b)(\alpha_2 + \alpha_3) \\
a_{22} &= v\rho(a+b)(\alpha_2 - \alpha_3) + \alpha_1 (c+d)(2\pi)^{-\frac{1}{2}} - u(c+d)(\alpha_2\beta + \alpha_3/\beta) \\
&\quad + \eta T(a+b)(1+\eta)^{-1} \\
b_1 &= b - \tfrac{1}{2}(\mathrm{erf}(T\beta(1+\eta)^{-\frac{1}{2}}) + \mathrm{erf}(T\beta^{-1}(1+\eta)^{-\frac{1}{2}})) \\
b_2 &= d - \tfrac{1}{2}(\mathrm{erfc}(T\beta(1+\eta)^{-\frac{1}{2}}) + \mathrm{erfc}(T\beta^{-1}(1+\eta)^{-\frac{1}{2}}))
\end{aligned}$$

$$\beta = (1-\rho)^{\frac{1}{2}}(1+\rho)^{-\frac{1}{2}} , \quad u = \tfrac{1}{2}(2\pi(1+\eta))^{-\frac{1}{2}} , \quad v = u(1-\rho^2)^{-\frac{1}{2}},$$

$$\alpha_1 = \exp(-T^2/2), \quad \alpha_2 = \exp(-T^2\beta^2/2(1+\eta)), \quad \alpha_3 = \exp(-T^2/2\beta^2(1+\eta)) .$$

Performance Evaluation:

The optimization criterion used in the threshold evaluation is the minimization of the probability of bit error. Therefore, the bit-error rate (BER) using both the detection scheme with and without memory should be compared. The probability of error expressions are given by

$$P(e)_{nm} = b + c \quad , \tag{17}$$

$$\begin{aligned}
P(e)_m &= (a+b)\mathrm{erfc}(Q) + (c+d)\mathrm{erfc}(R) + (a+\tfrac{1}{2})\mathrm{erf}(Q(1+\eta)^{-\frac{1}{2}}) \\
&\quad + c\ \mathrm{erf}(R(1+\eta)^{-\frac{1}{2}}) \\
&\quad + \tfrac{1}{2}\int_{u=Q}^{R} (2\pi(1+\eta))^{-\frac{1}{2}}\exp(-u^2/2)[\mathrm{erfc}\{(T-\rho u)((1+\eta)(1-\rho^2))^{-\frac{1}{2}}\} \\
&\qquad + \mathrm{erfc}\{(T+\rho u)((1+\eta)(1-\rho^2))^{-\frac{1}{2}}\}]\,du \quad .
\end{aligned} \tag{18}$$

It is observed that the probabilities of error for the two scheme are equal if $Q = R = T$, and it can be shown that $P(e)_m < P(e)_{nm}$ otherwise, if $r \neq 0$.

Results:

Numerical results were obtained for different values of the signal-to-noise ratio η. The numerical values of the thresholds and the probabilities of error are presented in Table 1. The correlation coefficient r for this set of numbers is one.

Table 1

η	T	Q	R	$P(e)_{nm}$	$P(e)_{m}$
10	1.6241	1.5307	1.7403	.2399	.2391
10^2	2.1590	2.0058	2.3152	.1004	.0994
10^3	2.6298	2.4576	2.7857	.0374	.0369
10^4	3.0350	2.8876	3.1802	.0133	.0131

VI. SUMMARY AND CONCLUSIONS

In this paper, a receiver with one-bit memory for channels which exhibit memory is developed. The dependence in the received data bits introduced by the fading in the channel is utilized to derive a receiver structure which yields a better performance than the standard memoryless receiver. An optimal solution is discussed, the complexity of implementation of the optimal scheme leads to the consideration of a suboptimal receiver. The suboptimal decision feedback algorithm employs two different thresholds depending upon the previous decision. A binary on-off keying system is considered as an example. The suboptimal receiver with one-bit memory is obtained and is shown to perform better than a memoryless receiver. The improvement in the probability of error, for the example considered, is not large due to the nature of the problem. The objective was to exhibit the theoretical feasibility of the idea and it is expected that such a receiver would attain significant improvement when used in a real communication environment. This idea could be applied to other fading channels as well for example to channels which exhibit phase fading or amplitude and phase fading both. It is also expected that a detection algorithm with larger memory will perform even better but it would also result in a more complex implementation.

VII. REFERENCES

1. R. W. Chang and J. C. Hancock, "On receiver structures for channels having memory," IEEE Trans. Inform. Theory, vol. IT-12, pp. 463-468, Oct. 1966.
2. K. Abend and B. D. Fritchman, "Statistical detection for communication channels with intersymbol interference," Proc. IEEE, vol. 58, pp. 779-785, May 1970.
3. T. M. Cover, "Hypothesis testing with finite statistics," Ann. Math. Statist., vol. 40, pp. 828-835, 1969.
4. R. A. Roberts and J. R. Tooley, "Estimation with finite memory," IEEE Trans. Inform. Theory, vol. IT-6, pp. 685-691, Nov. 1970.
5. E. G. Baxa and L. W. Nolte, "Adaptive signal detection with finite memory," IEEE Trans. System, Man, Cybern., vol. SMC-2, pp. 42-49, Jan. 1972.
6. J. Koplowitz and R. A. Roberts, "Sequential estimation with a finite statistic," IEEE Trans. Inform. Theory, vol. IT-19, pp. 631-636, Sept. 1973.
7. C. T. Mullis and R. A. Roberts, "Finite-memory problems and algorithms," IEEE Trans. Inform. Theory, vol. IT-20, pp. 440-455, July 1974.
8. H. L. Van Trees, Detection, Modulation and Estimation Theory; Part I. New York, Wiley, 1968.

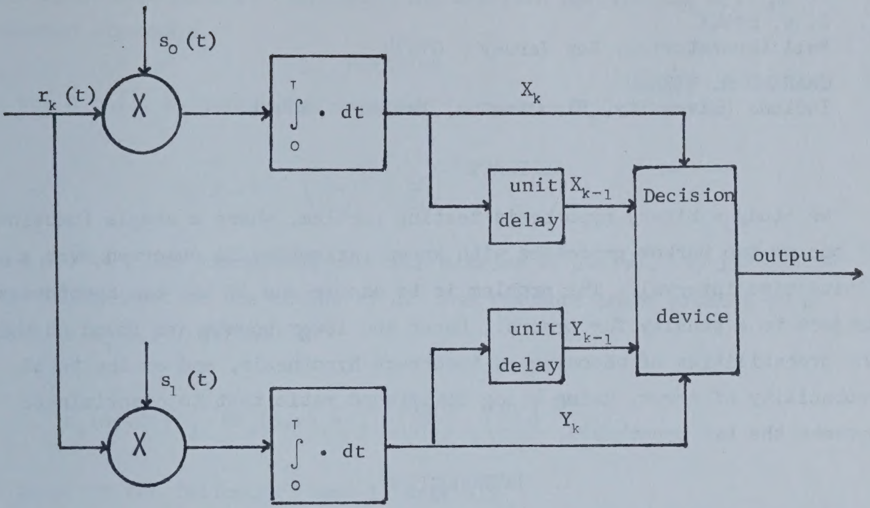

$s_0(t)$
$r_k(t)$
X
$\int_0^T \cdot\, dt$
X_k
unit delay
X_{k-1}
Decision device
output
unit delay
Y_{k-1}
$s_1(t)$
X
$\int_0^T \cdot\, dt$
Y_k

Fig. 1. Proposed optimum receiver structure.

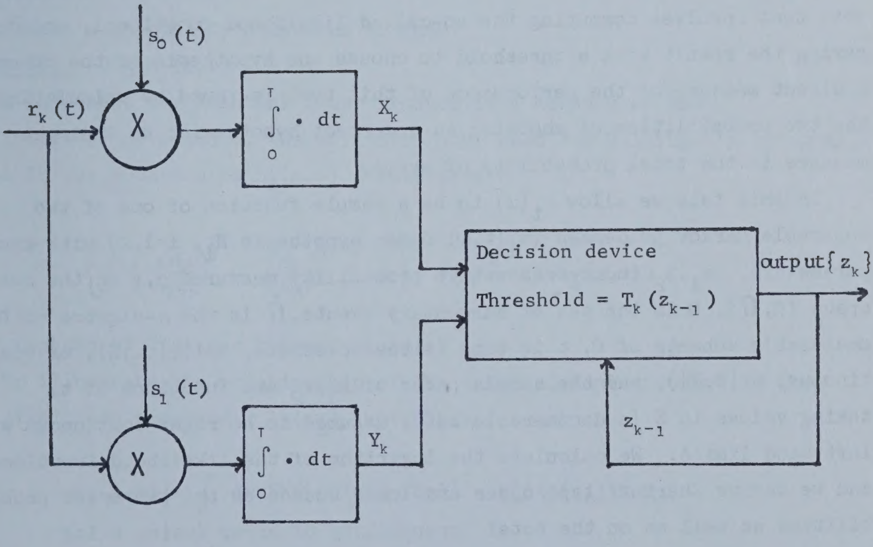

$s_0(t)$
$r_k(t)$
X
$\int_0^T \cdot\, dt$
X_k
Decision device
Threshold = $T_k(z_{k-1})$
output$\{z_k\}$
z_{k-1}
$s_1(t)$
X
$\int_0^T \cdot\, dt$
Y_k

Fig. 2. Suboptimum detector structure.

LIKELIHOOD FUNCTIONALS AND CHERNOFF ERROR PROBABILITY BOUNDS FOR DISCRIMINATING BETWEEN MARKOV PROCESSES

B. W. STUCK
Bell Laboratories, New Jersey 07974

CHARLES M. NEWMAN
Indiana University, Bloomington, Indiana 47401

ABSTRACT

We study a binary hypothesis testing problem, where a sample function of one of two Markov processes with known parameters is observed over a finite time interval. The problem is to choose one of the two hypotheses, subject to a penalty for errors. Upper and lower bounds are found on both the probabilities of choosing an incorrect hypothesis, and on the total probability of error, using a log likelihood ratio test to discriminate between the two hypotheses.

1. INTRODUCTION

A great deal of the mathematical statistics literature is concerned with classifying a sample function $r_t(\omega)$ of a random process observed over a finite time interval as coming from one of two classes, called hypotheses. The test for discriminating between hypotheses is chosen to minimize a cost functional, which penalizes errors. For a wide variety of cost functionals this test involves computing the so-called likelihood functional, and comparing the result with a threshold to choose one hypothesis or the other. A direct measure of the performance of this test is found by calculating the two probabilities of choosing an incorrect hypothesis; an indirect measure is the total probability of error.

In this talk we allow $r_t(\omega)$ to be a sample function of one of two separable Markov processes ($x_i(t,\omega)$ under hypothesis H_i, $i=1,2$) with known parameters. x_1, x_2 induce respective probability measures ρ, ν on the measure space $(\Omega, \mathcal{G})$, Ω is the set of elementary events, $\mathcal{G}$ is the σ-algebra of Borel measurable subsets of Ω, t is time (either discrete, $t \in \{0,\ldots,N\}$, or continuous, $t \in [0,T]$), and the sample paths of x_1, x_2 are functions of t, taking values in S (a denumerable set), assumed to be right continuous with left hand limits. We calculate the logarithm of the likelihood functional, and we derive Chernoff type upper and lower bounds on the two error probabilities as well as on the total probability of error (using a log likelihood ratio test). Our principal tool in calculating these bounds is the Kakutani inner product of ρ with ν, which also allows us to obtain new conditions for ρ and ν to be orthogonal.

Time permitting, the extension of these results to the continuous state space case will be sketched. The detailed derivations will be presented elsewhere.

2. NOTATION

For $0 < s < 1$, define the nonnegative measure $H_s(\rho,\nu)$,

$$H_s(\rho,\nu) = \left(\frac{d\rho}{d\mu}\right)^s \left(\frac{d\nu}{d\mu}\right)^{1-s} \mu$$

where $\rho,\nu \ll \mu$ (an example of one such measure is $\mu=\rho+\nu$). By inspection, H_s is independent of the choice of μ. The Kakutani inner product of ρ with ν is $h_s(\rho,\nu)$, defined as

$$h_s(\rho,\nu) = \int dH_s(\rho,\nu) = \int \left(\frac{d\rho}{d\nu}\right)^s \left(\frac{d\nu}{d\mu}\right)^{1-s} d\mu \ .$$

The proof of the following lemma is trivial:

Lemma 2.1. $\rho \perp \nu$ iff $H_s(\rho,\nu) = 0$ iff $h_s(\rho,\nu) = 0$, for some $s \in (0,1)$.

Using Hölder's inequality it is straightforward to show:

Lemma 2.2. $h_s(\rho,\nu)$ and its logarithm are convex functions of s. Moreover, the convexity is strict iff $(d\rho/d\nu)(x)$ is not constant for all $x \in \text{supp}(\rho) \cap \text{supp}(\nu)$.

Jensen's inequality allows us to show

Lemma 2.3. $0 \leq h_s(\rho,\nu) \leq 1$

which indicates the Kakutani inner product is a measure of the "colinearity" of ρ and ν. We will also have need for Hellinger's integral, which in our notation is $h_{\frac{1}{2}}(\rho,\nu)$. Note that

$$\left\{\int \left(\left(\frac{d\rho}{d\mu}\right)^{\frac{1}{2}} - \left(\frac{d\nu}{d\mu}\right)^{\frac{1}{2}}\right)^2 d\mu\right\}^{\frac{1}{2}} = \sqrt{2}\ \left(1-h_{\frac{1}{2}}(\rho,\nu)\right)$$

is a measure of distance between ρ and ν.

To illustrate the relationship of the Kakutani inner product to the log likelihood functional $\Lambda(r_t)$, note that

$$h_s(\rho,\nu) = \int d\nu \ \text{EXP}[s \ln(d\rho/d\nu)] = E[\text{EXP}(s\Lambda) \mid H_2]$$

$$h_s(\rho,\nu) = \int d\rho \ \text{EXP}[(s-1)\ln(d\rho/d\nu)] = E[\text{EXP}(s-1)\Lambda) \mid H_1]$$

which are simply the Laplace transforms of the log likelihood distribution, evaluated at s and (s-1), respectively, assuming H_2 and H_1 are respectively true.

H. Chernoff [1] exploited this relationship to show

$$\Pr[\text{choose } H_2 \mid H_1 \text{ true}] \leq \underset{s\leq 1}{\text{INF}}\; h_s(\rho,\nu)e^{-(s-1)L},$$

$$\Pr[\text{choose } H_1 \mid H_2 \text{ true}] \leq \underset{s\geq 0}{\text{INF}}\; h_s(\rho,\nu)e^{-sL},$$

where L is the threshold in the log likelihood ratio test. Shannon, Gallager and Berlekamp [2] obtained similar lower bounds on these two probabilities. Using Hellinger's integral, upper and lower bounds on the total probability of error can be found [3,4]:

$$\tfrac{1}{2}\min(\pi_1,\pi_2)h^2_{\frac{1}{2}}(\rho,\nu) \leq P_E \leq (\pi_1\pi_2)^{\frac{1}{2}}\, h_{\frac{1}{2}}$$

$$P_E = \pi_1\cdot\Pr[\text{choose } H_2 \mid H_1 \text{ true}] + \pi_2\cdot\Pr[\text{choose } H_1 \mid H_2 \text{ true}]$$

where π_j is the apriori probability H_j is true.

3. DISCRETE TIME, FINITE STATE MARKOV PROCESSES

In this section x_1,x_2 are two time homogeneous Markov processes taking values in a finite state space $S = \{1,\ldots,M\}$ at discrete time instants $t \in \{0,\ldots,N\}$. Let u,v be the respective initial distribution vectors of x_1,x_2, so $u_i,v_i \geq 0$ for $1 \leq i \leq M$ and $u_1+\ldots+u_M = v_1+\ldots+v_M = 1$. To avoid trivialities the Kakutani inner product of u with v is assumed nonzero,

$$h_s(u,v) = \sum_{i=1}^{M} u_i^s\, v_i^{1-s} = H_s^{TR}(u,v)e$$

$$H_s(u,v)_i = u_i^s\, v_i^{1-s};\quad e_i = 1,\quad 1 \leq i \leq M.$$

Denote the transition matrices of x_1,x_2 by P,Q

$$P_{ij} = \Pr[x_1(k+1)=j \mid x_1(k)=i] \qquad 1 \leq i,\ \ j \leq M$$

$$Q_{ij} = \Pr[x_2(k+1)=j \mid x_2(k)=i] \qquad 0 \leq k \leq N-1$$

$$\sum_{j=1}^{M} P_{ij} = \sum_{j=1}^{M} Q_{ij} = 1 .$$

Define the M×M matrix $H_s(P,Q)$ by

$$H_s(P,Q)_{ij} = P_{ij}^{s} \; Q_{ij}^{1-s} \; , \quad 1 \leq i, \; j \leq M$$

Proposition 3.1. Given the above,

a) $$\Lambda(r_0,\dots,r_N) = \ln \frac{u_{j0}}{v_{j0}} + \sum_{k=0}^{N-1} \ln\left(P_{j_k j_{k+1}} / Q_{j_k j_{k+1}}\right)$$

where $j_k (1 \leq j_k \leq M)$ is the state of r_k, $0 \leq k \leq N$.

b) $$h_s(\rho,\nu) = H_s^{TR}(u,v)\cdot[H_s(P,Q)]^N \cdot e$$

We now relate the asymptotic behavior (large N) of $h_s(\rho,\nu)$ to an eigenvalue of $H_s(P,Q)$. We do so using standard results from operator theory, so the generalization of our results from the countable state space case to the continuous state space case will be immediate. Consider $H_s(P,Q)$ as an operator acting on a Banach space B (taken here to be ℓ_∞). The resolvent of $H_s(P,Q)$ is defined to be the bounded linear operator $(\lambda I - H_s)^{-1}$, wherever it exists. The set of λ where the resolvent exists is called the resolvent set, and the complement of the resolvent set is called the spectrum of the operator, $\sigma(H_s)$, which is closed and bounded. Define the spectral radius of H_s as

$$r(H_s(P,Q)) = \sup_{\lambda \in \sigma(H_s)} |\lambda|$$

Lemma 3.2. $\limsup_{N\to\infty} \frac{1}{N} \ln[h_s(\rho,\nu)] \leq \ln r(H_s(P,Q))$

Proposition 3.3. There exists at least one eigenvector w of $H_s(P,Q)$ with nonnegative entries and with eigenvalue $r(H_s)$. If $w^{TR}\cdot H_s(u,v) \neq 0$, then

$$\lim_{N\to\infty} \frac{1}{N} \ln\,[h_s(\rho,\nu)] = \ln r(H_s(P,Q))$$

4. CONTINUOUS TIME, FINITE STATE MARKOV PROCESSES

In this section x_1, x_2 are two time homogeneous Markov processes taking values in a finite state space $S = \{1,\dots,M\}$ at time $t \in [0,T]$. Let u,v be the respective initial distribution vectors of x_1, x_2, and to avoid trivialities assume u and v are not mutually orthogonal.

The transition probabilities of x_1, x_2 are given by

$$P(t) = \mathrm{EXP}(tA), \quad Q(t) = \mathrm{EXP}(tB)$$

$$P_{ij}(t) = \Pr[x_1(s+t)=j \mid x_1(s)=i] \qquad s, s+t \in [0,T]$$

$$Q_{ij}(t) = \Pr[x_2(s+t)=j \mid x_2(s)=i] \qquad 1 \leq i, \;\; j \leq M$$

where A,B are the infinitesimal generators of P(t), Q(t). The elements A_{ij}, B_{ij} are assumed finite, and correspond to the rates at which x_1, x_2 move from state i to state j.

Let $A = A_d + A_0$, $B = B_d + B_0$, where A_d, B_d are diagonal matrices with non-positive entries, and A_0, B_0 have all zero diagonal entries with non-negative off diagonal elements.

Proposition 4.1.

a) $$\Lambda(r_t, t \in [0,T]) = \ln \frac{u_{j0}}{v_{j0}} + \int_0^T \left[\left(A_{j_k j_k} - B_{j_k j_k} \right) + \sum_k \ln \left(A_{j_k j_{k+1}} / B_{j_k j_{k+1}} \right) \delta(t-t_k) \right] dt$$

where $\{t_k\}$ are the set of time instants r_t changes state, and $\{j_0, j_1, \ldots\}$ label the successive state of r_t.

b) $$h_s(\rho, \nu) = H_s^{TR}(u,v) \cdot \mathrm{EXP}[T(sA_d + (1-s)B_d + H_s(A_0, B_0))] \cdot e \,.$$

Define $C_{ij} = A_{ij}^{s} B_{ij}^{1-s}$ $(i \neq j)$, $C_{ii} = - \sum\limits_{k \neq i} A_{ik}^{s} B_{ik}^{1-s}$, $D_{ij} = 0$ $(i \neq j)$,

$D_{ii} = C_{ii} - sA_{ii} - (1-s)B_{ii} \geq 0$. Then

$$h_s(\rho, \nu) = H_s^{TR}(u,v) \cdot \mathrm{EXP}[T(C-D)] \cdot e$$

The Kakutani inner product can be rewritten as the expectation of a functional of a new Markov process:

Proposition 4.2. Let $x_3(t)$ be a finite state continuous time Markov process, with state space $S = \{1, \ldots, M\}$, $t \in [0,T]$, transition probabilities

$R(t) = \mathrm{EXP}(tC)$,

$$R_{ij}(t) = \Pr[x_3(s+t)=j \mid x_3(s)=i], \quad 1 \leq i, \quad j \leq M$$

and initial distribution $H_s(u,v)/h_s(u,v)$. Then

$$h_s(\rho,\nu) = h_s(u,v) E_{x_3}\left[\mathrm{EXP}\left(-\int_0^T D(x_3(t))dt\right)\right]$$

where $D(x) = D_{xx}$.

The Kakutani inner product depends asymptotically (large T) on the spectral radius of (C-D):

Proposition 4.3. $\underset{T\to\infty}{\mathrm{LIM\ SUP}} \ \frac{1}{T} \ln[h_s(\rho,\nu)] \leq r(C-D)$.

Proposition 4.4. If there exists at least one eigenvector w with eigenvalue r(C-D) of (C-D), $w_i > 0$ for all i, and if $w^{TR} \cdot H_s(u,v) \neq 0$

$$\underset{T\to\infty}{\mathrm{LIM}} \ \frac{1}{T} \ln[h_s(\rho,\nu)] = r(C-D).$$

REFERENCES

1) H. Chernoff, "A Measure of Asymptotic Efficiency For Tests of a Hypothesis Based on a Sum of Observations," Ann. Math. Stat., 23, 493-507, (1952).

2) C. E. Shannon, R. G. Gallager, E. R. Berlekamp, "Lower Bounds to Error Probability For Coding on Discrete Memoryless Channels, 1" Info. Con., 10, 65-103 (1967).

3) C. H. Kraft, "Some Conditions For Consistency and Uniform Consistency of Statistical Procedures," Univ. Cal. (Berkeley) Pub. Statistics, 2(6), 125-142) (1955).

4) T. T. Kadota, L. A. Shepp, "On the Best Set of Linear Observables For Discriminating Two Gaussian Signals," IEEE Trans. Info. Thy., 13, 278-284 (1967).

HOW TO TRACK A SWARM OF FIREFLIES BY OBSERVING THEIR FLASHES* **

DONALD L. SNYDER AND PHILIP M. FISHMAN
Biomedical Computer Laboratory
Washington University School of Medicine
St. Louis, Missouri 63110

ABSTRACT

We examine the problem of estimating the centroid of a normally shaped intensity function of a time-space point process in terms of an observed realization of the point process. The centroid is assumed to move randomly as a projection of a Gauss-Markov process. We find that the centroid is conditionally normal given the observations. Dynamical equations are given for the conditional mean and covariance of the centroid. The resulting filter is nonlinear but closely related to the continuous-discrete Kalman-Bucy filter. An application to optical position sensing is given.

---

* This work was supported by the National Science Foundation under Grant ENG74-07800 and by the National Institutes of Health under Research Grant RR00396 from the Division of Research Resources.

** Published in the IEEE Transactions on Information Theory, November 1975.

AN APPROXIMATION FOR THE AMPLITUDE DISTRIBUTION OF SHOT NOISE

O. YUE, R. LUGANNANI, AND S.O. RICE
Department of Applied Physics and Information Science
University of California, San Diego
La Jolla, California 92093

ABSTRACT

An approximate method for evaluating the amplitude density of shot noise is presented. The approximation consists of a sum of Gaussian densities; the parameters of the approximation are chosen by matching its moments with those of the original shot process. Accuracy and ease of computation are the outstanding features of this method. Comparisons with exact calculations in two cases of practical interest show good agreement for a wide range of values.

INTRODUCTION

The classical shot noise process can be characterized by

$$y(t) = \sum_i f(t-t_i) \qquad (1)$$

where the t_i's are Poisson distributed with constant rate ν and $f(t)$ is a nonrandom function specifying the pulse shape. This model can be generalized by letting $f(t-t_i) = a_i f(t,t_i)$ where the amplitude a_i is random and/or allowing the underlying Poisson process to be non-stationary. Applications of this type of process can be found in radar [1], underwater communication [2], geological surveys [3], aircraft and bridge design [4], telephone systems [5] and reliability prediction [6].

The probability density function of the amplitude of the process (1) is given formally by the inversion integral [7]

$$p(y) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \Phi(iu) \cdot \exp\,[-iuy]du \qquad (2)$$

where

$$\Phi(iu) = E \exp\,[iuy] = \exp\left[\nu\int_{-\infty}^{\infty}\left\{\exp[iuf(t)]-1\right\}dt\right]. \qquad (3)$$

Note that the process y(t) is stationary so that its density is not a function of time. Much research effort has been directed toward the evaluation of the above inversion integral and its more general forms for several pulse shapes of interest. Rice considered the use of an Edgeworth Series approximation [7], and Gilbert and Pollack [8] derived an integral equation whose solution yields the desired density. Janssen and Lambert simplified the integral for very low repetition rates, ν, and found good agreement with Monte Carlo simulations [9]. An approximation method using the saddle-point approach was reported by Roberts [10] who used a quadratic approximation to find the saddle-point location; he reported good accuracy for large values of ν. More recently Richter and Smits [11] described a generalized approach for the numerical evaluation of the integral. For the most part, these methods have been concerned with special cases and are either too time consuming for computation or lack the desired accuracy for wide-spread general application.

In this paper we present an approximate method for evaluating the shot noise amplitude density, whose advantages include ease of computation and good accuracy for a wide range of values of ν. We shall restrict ourselves to pulses that vanish outside the interval [0,T] and that have finite energy; these restrictions permit us to concentrate on the salient features of the approximation. They are satisfied in many practical situations.

It is convenient to define

$$H(iu) = \frac{1}{T}\int_0^T \exp\,[iuf(t)]dt \tag{4}$$

where H(iu) is itself a characteristic function whose associated density will be denoted by $p_1(y)$. This density may be viewed as arising from the transformation $y = f(t)$ where t is uniformly distributed on the interval [0,T]. Since f(t) has duration T, we may write (3) as

$$\Phi(iu) = \exp\,[\nu T(H(iu) - 1)]. \tag{5}$$

Then p(y) can be expressed formally as the infinite sum

$$\begin{aligned} p(y) &= \exp\,[-\nu T]\sum_{k=0}^{\infty}\frac{(\nu T)^k}{k!}\,\frac{1}{2\pi}\int_{-\infty}^{\infty} H^k(iu)\,\exp\,[-iuy]du \\ &= \exp\,[-\nu T]\sum_{k=0}^{\infty}\frac{(\nu T)^k}{k!}\,p_k(y) \end{aligned} \tag{6}$$

where $p_k(y)$ is the k-th convolution of $p_1(y)$ with itself. The function $p_k(y)$ can be regarded as the probability density of the sum of k independent random variables each having probability density $p_1(y)$. As a notational convenience, we adopt the convention $p_0(y) = \delta(y)$. In general the $p_k(y)$'s are not expressible in a simple closed form. However, for certain special choices of $p_1(y)$, the sum in (6) can be readily evaluated.

The approach used in this paper is to approximate $p_1(y)$ by a sum of Gaussian densities, whose convolutions are easily accomplished. These convolutions are then used in the sum (6) to obtain an approximate expression for p(y).

THE APPROXIMATION

The most important criterion for choosing an approximation for $p_1(y)$ is that convolutions with itself be expressible in terms of simple, tractable functions. One simple example of an approximation is the single point density $\delta(y-a)$, in which case expression (6) becomes the Poisson density. Other discrete densities that satisfy this criterion include the binomial and Poisson densities.

A brief survey of continuous densities yields three candidates: the Gaussian, Cauchy, and Gamma densities. The Cauchy density has infinite moments of all orders, and since most processes encountered in practice have finite moments, the Cauchy density does not appear to be a good choice for an approximation. Since the Gamma density is defined only for positive values its usefulness is somewhat limited and it too is not a good choice. This leaves the Gaussian density as the most promising candidate to use in the approximation.

It is well known that a Gaussian density reproduces itself upon convolution, and this property can also be shown to hold for a sum of Gaussian densities with different means and standard deviations. The freedom to choose the various means and standard deviations as well as the number of terms in the sum gives us the ability to approximate a rather large class of densities, $p_1(y)$. Accordingly the general form of the approximating function we shall use is

$$\sum_{\ell=1}^{L} w_\ell N(y;\mu_\ell, \sigma_\ell) \tag{7}$$

where $N(y;\mu,\sigma) = (2\pi\sigma^2)^{-\frac{1}{2}} \cdot \exp[-(y-\mu)^2/2\sigma^2]$ and $w_1 + w_2 + \cdots + w_L = 1$.

To determine the parameters w_ℓ, μ_ℓ, and σ_ℓ we consider the corresponding approximation to the characteristic function

$$H(iu) = \int_{-\infty}^{\infty} \exp[iuy]p_1(y)dy \cong \sum_{\ell=1}^{L} w_\ell \exp[iu\mu_\ell - \tfrac{1}{2}u^2\sigma_\ell{}^2]. \tag{8}$$

In general, it is not possible to achieve good agreement over the entire range of u using small values of L. However, in many applications it is the small values of u that are most important and we accordingly shall try to approximate H(iu) most accurately in that region.

To demonstrate the importance of the small values of u in evaluating the density, we present the following heuristic argument; the results are of some independent interest in the study of the behavior of p(y) for large arguments. We begin by considering the inversion integral

$$p(y) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \exp[-iuy + \nu T(H(iu)-1)]du = \frac{1}{2\pi}\int_{-\infty}^{\infty} \exp[G(iu)]du.$$

For most pulses encountered in practice there is a unique saddle point of G(iu) located on the imaginary u axis whenever y belongs to the domain of the density p(y) [12]. Since G(iu) is analytic in the complex u-plane we may deform the path of integration to pass through the imaginary axis saddle point. Usually the contribution to the integral from this saddle point provides a good approximation to p(y) when y is large. This approximation is given by

$$p(y) \sim (-2\pi G^{(2)}(iu_o))^{-\frac{1}{2}} \cdot \exp[G(iu_o)] \tag{9}$$

where u_o is the imaginary axis saddle point and $G^{(2)}(iu_o)$ is the second derivative of G(iu) evaluated at $u = u_o$.

The saddle point is found by solving the equation $G^{(1)}(iu_o) = 0$ or, equivalently,

$$y = \nu \int_0^T f(t) \cdot \exp\,[b_o f(t)]dt$$

where we have made the substitution $b_o = iu_o$. Assuming that f(t) has a single maximum in the interval [0,T], denoted by f_o, we may write another asymptotic expression for the above integral which appears as [13]

$$y \sim \nu f_o(-2\pi/b_o f^{(2)}(t_o))^{\frac{1}{2}} \cdot \exp\,[b_o f_o]$$

with $f^{(2)}(t_o)$ the second derivative of f(t) evaluated at $t = t_o$. From this we obtain an asymptotic expression relating b_o and y

$$b_o \sim f_o^{-1} \ln y + \text{correction terms}. \tag{10}$$

Finally, by combining (9) and (10) we obtain an approximation for p(y) when y is large

$$p(y) \sim (2\pi f_o y)^{-\frac{1}{2}} \cdot \exp\,[-f_o^{-1} y \ln y\,] + \text{correction terms} \tag{11}$$

From (10) it is evident that even for relatively large values of y, the region of interest for the saddle point b_o is near the origin. Since this saddle point plays a major role in determining the density p(y), the value of H(iu) for small arguments is clearly of great importance for pulses satisfying these assumptions.

While the above analysis applies to a pulse satisfying a specific set of assumptions, our experience has shown that similar results hold for many other pulses. For this reason we will choose the parameters in the approximation (8) to obtain good agreement near the origin.

The coefficients in the power series representation of H(iu) are simply related to the moments of $p_1(y)$ and we have

$$H(iu) = \sum_{k=0}^{\infty} \frac{\alpha_k}{k!}\,(iu)^k$$

where

$$\alpha_k = \int_{-\infty}^{\infty} y^k p_1(y)dy = \frac{1}{T}\int_0^T f^k(t)dt.$$

Similarly for the approximating characteristic function

$$\sum_{\ell=0}^{L} w_\ell \, \exp[iu\mu_\ell - \tfrac{1}{2}u^2\sigma_\ell^2\,] = \sum_{k=0}^{\infty} \frac{\beta_k}{k!}(iu)^k$$

and

$$\beta_k = \sum_{\ell=1}^{L} w_\ell \sum_{\substack{p=0\\ p\text{-even}}}^{k} \binom{k}{p}\, \sigma_\ell^{\,p} \mu_\ell^{\,k-p} (p-1)(p-3)\cdots(1).$$

Therefore, to obtain the best fit at the origin we shall equate the first 3L-1 moments of the two characteristic characteristic functions. That is, we will set $\beta_k = \alpha_k$ for $k = 0,1,\cdots, 3L-1$ and solve for w_ℓ, μ_ℓ, and σ_ℓ for $\ell = 1,2,\cdots, L$. (There are L values of each of the parameters w_ℓ, μ_ℓ, and σ_ℓ, requiring a total of 3L equations; matching the zeroth moment ensures that $w_1 + w_2 + \cdots + w_L = 1$.) If $p_1(y)$ has symmetry properties, the number of equations to be solved is often significantly reduced. Experience has shown that if better agreement to p(y) is desired for very large values of y, then some of the higher order moments of the two characteristic functions should be equated rather than the first 3L-1.

In Middleton's work [14], the pulse was assumed to be a narrowband signal and the exponent of the characteristic function contained a Bessel function, which was approximated with a single Gaussian density by matching the variance. For this case L = 1 and (6) becomes

$$p(y) \cong \exp[-\nu\tau]\sum_{k=0}^{\infty} \frac{(\nu T)^k}{k} (2\pi k\sigma^2)^{-\frac{1}{2}} \exp[-(y-k\mu)^2/2k\sigma^2]. \tag{12}$$

When νT is large this approximation can be justified by a heuristic argument based upon the central limit theorem. As νT increases each $p_k(y)$, being the convolution of $p_1(y)$ with itself k times, can be approximated by a Gaussian density. This Gaussian approximation is quite good within a few standard deviations of the mean even for only moderately large values of k. Thus (6) consists of a sum of nearly Gaussian terms, and it is not too surprising that (12) provides a good approximation.

However, for small values of νT and for other pulse shapes a single term in the approximation (7) is usually not adequate. As L becomes large, more parameters are available to provide a better approximation to $p_1(y)$; unfortunately, the computational complexity increases rapidly as L increases. The problem is essentially one of determining the coefficients of an L-th order polynomial raised to the k-th power. We have restricted our investigation in this paper to the case where L = 2. We have found that this case provides a good fit for many shot processes. Two examples illustrating this are considered in the following section.

EXAMPLES

The primary purpose of the foregoing development is to obtain good, tractable approximations for the shot noise densities. To illustrate the accuracy of our approximation, two pulse shapes will be considered. These are:

$$f_1(t) = \begin{cases} \sin \omega t, & 0 \le t \le \frac{2\pi}{\omega} = T \\ 0 \quad , & \text{otherwise} \end{cases}$$

$$f_2(t) = \begin{cases} 1 - \cos \omega t, & 0 \le t \le \frac{2\pi}{\omega} = T \\ 0 \quad , & \text{otherwise} \end{cases}$$

We will compare the exact densities determined by these two pulses with the appropriate approximation determined as above. Although the difference between $f_1(t)$ and $f_2(t)$ is a shift in amplitude, the reader is cautioned not to jump to the conclusion that a shift in the mean of $p_1(y)$ results simply in a shift in the mean of p(y). In fact since the maximum amplitude of

$f_2(t)$ is twice that of $f_1(t)$, the asymptotic expression (11) shows that, for large y, the density determined by $f_1(t)$ decays twice as fast as the corresponding density determined by $f_2(t)$.

For both pulses the corresponding density, p(y), has singularities at various values of y. Consequently, straightforward application of general Fourier inversion techniques such as the Fast Fourier Transform to the integral (2) will not generally give good results. An efficient numerical method is described in [16] that can be used to determine the exact values for p(y). This method subtracts an asymptotic component from the integrand so that the remaining difference can be evaluated with relative ease using the trapezoidal rule. The subtracted terms are expressible in closed form and must, of course, be added to the result of the numerical integration.

For the sine pulse, $H(iu) = J_o(u)$ with

$$p_1(y) = \begin{cases} (\pi\sqrt{1-y^2})^{-1}, & |y| \le 1 \\ 0 & , \text{otherwise} \end{cases}$$

The first convolution of $p_1(y)$ with itself is

$$p_2(y) = \int_{-\infty}^{\infty} p_1(x)p_1(y-x)dx = \begin{cases} K(m)/\pi^2, \ m = 1 - \frac{y^2}{4}, & |y| \le 2 \\ 0 & , \text{otherwise} \end{cases}$$

where K(m) is a complete elliptic integral of the first kind [15]. However, for $k>2$, $p_k(y)$ cannot be expressed in closed form, and the exact expression is computed numerically as described above. Besides the unit impulse at the origin, which is common to all shot processes whose pulses have finite duration, the density associated with $f_1(t)$ becomes infinite at $y = 0$ due to $p_2(y)$ and at $y = \pm 1$ due to $p_1(y)$.

The parameters used in the approximation of $p_1(y)$ are $w_1 = w_2 = 0.5$, $\sigma_1{}^2 = \sigma_2{}^2 = 0.06699$, $\mu_1 = 0.6580$, and $\mu_2 = -0.6580$.

The exact probability densities together with the approximate densities are shown in Figure 1 for values of $\nu T = 0.1$, 1.0, and 10.0. Since the densities are symmetric, they are plotted only for positive values of the argument. It follows from (6) that the contribution by $p_k(y)$ is weighted by $(\nu T)^k/k!$ so that for $\nu T = 0.1$ p(y) is dominated by the first few terms of the sum. The result is a staircase-like density function. The discontinuity at $y = 2$ is due to $p_2(y)$ which approaches a non-zero value as $y \uparrow 2$ and vanishes for $y>2$. Pronounced discontinuties in the derivatives persist throughout. For $\nu T = 1.0$, p(y) exhibits similar behavior, but becomes smoother much sooner than before. For $\nu T = 10.0$ the probability of pulses overlapping is large and the resulting density becomes Gaussian-like because of the central limit theorem. The singularities at $y = 0$ and and $y = 1$ and the discontinuity at $y = 2$ are still present but have become much less pronounced.

For $\nu T = 0.1$, the approximate density peaks at about $y = 0.7$ and then after crossing p(y) several times, becomes tangential to it. The approximation is not very good for the range of values shown on the graph. However the approximation gives the rate of decay very accurately. When νT is increased to 1.0, the approximation is much better; for $3.5<y<7.5$, the relative error is within 3%. For $\nu T = 10.0$, the approximate density is not distinguishable from p(y); for $0<y\le 13.0$, the first two significant digits

agree, except around discontinuities.

Turning to the raised cosine pulse, we have $H(iu) = e^{iu}J_o(u)$ with

$$p_1(y) = \begin{cases} [\pi\sqrt{y(2-y)}]^{-1}, & 0 \le y \le 2 \\ 0 & \text{, otherwise} \end{cases}$$

$$p_2(y) = \begin{cases} K(m)/\pi^2, & m = y - y^2/4, 0 \le y \le 4 \\ 0 \quad , & \text{otherwise.} \end{cases}$$

In this case $p_1(y)$ is infinite at $y = 0$ and $y = 2$ and $p_2(y)$ becomes infinite at $y = 2$. The parameters used in approximating $p_1(y)$ are $w_1 = w_2 = 0.5$, $\sigma_1{}^2 = \sigma_2{}^2 = 0.06699$, $\mu_1 = 0.3420$, and $\mu_2 = 1.6580$. In other words the approximation is just a shifted version of the previous one.

The exact and approximate densities are shown in Figure 2 for the same values of νT as in the earlier case. One obvious difference between the exact densities in Figure 2 and those in Figure 1 is that the former vanish for $y<0$. Otherwise the behavior of the curves for $\nu T = 0.1$ and 1.0 is quite similar. For $\nu T = 10.0$, the density peaks at $y = 9.2$ which is very close to the mean of the shot process. As in Figure 1, $p(y)$ is very smooth and Gaussian-like when $\nu T = 10.0$.

For $\nu T = 0.1$, the approximate density follows the oscillations of $p(y)$ very closely. The relative error is less than 3% for $y>2$, except near the even integers. For $\nu T = 1.0$, the two curves overlap for most values of $y>4$. As before, the agreement for $\nu T = 10.0$ is within two to three significant digits.

CONCLUSION

In this paper we have presented an approximation for the first order density of shot noise by using a sum of gaussian densities to represent the pulse density $p_1(y)$. The parameters in this representation are determined by the moments of $p_1(y)$. Examples with the sine and raised cosine pulse demonstrated good accuracy in the region of greatest interest for a wide range of pulse rates, ν. In these examples we have used the first five moments of $p_1(y)$ to determine the approximations. We have also computed similar results for the cumulative distribution in these two cases and even better agreement was obtained because of the smoothing effect of the integration. In situations where the pulse shape not known, the first few moments can be derived from the semi-invariants of the shot process, which can be determined experimentally.

ACKNOWLEDGEMENT

This research was sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant No. AFOSR 74-2689.

REFERENCES

[1] H.L. VanTrees, Detection, Estimation and Modulation Theory, Part III, Wiley, New York (1971).

[2] D. Middleton, "A Statistical Theory of Reverberation and Similar First-Order Scattered Fields", IEEE Trans. Inf. Theory, Vol. 13, pp. 372-392, 1967.

[3] J.H. Ganton and A.R. Milne, "Temperature-and Wind-Dependent Ambient Noise Under Midwinter Pack Ice", J. Acoust. Soc. Am., Vol. 38, pp. 406-411, 1965.

[4] S.C. Liu, "Dynamics of Correlated Random Pulse Trains", J. of the Engineering Mechanics Division, ASCE, Vol. 96, pp. 455-470, 1970.

[5] J.H. Fennick, "A Report on Some Characteristics of Impulse Noise in Telephone Communication", IEEE Trans. Commun. Electron, Vol. 83, pp. 700-705, 1964.

[6] T.I. Smits and R.F. Lambert, "System-Reliability Prediction in Impulse-Noise Environments, Based on Wear-Dependent Failure Rates, Using Response 'Peak' Statistics", J. Acoust. Soc. Am., Vol. 43, pp. 1344-1350, 1968.

[7] S.O. Rice, "Mathematical Analysis of Random Noise", Bell Syst. Tech. J., Vol. 23, pp. 282-332, 1944.

[8] E.N. Gilbert and H.O. Pollak, "Amplitude Distribution of Shot Noise", Bell Syst. Tech. J., Vol. 39, pp. 333-350, 1960.

[9] R.A. Janssen andR.F. Lambert, "Numerical Calculation of Some Response Statistics for a Linear Oscillator Under Impulse-Noise Excitation", J. Acoust. Soc. Am., Vol. 41, pp. 827-835, 1967.

[10] J.B. Roberts, "Distribution of the Responses of Linear Systems to Poisson Distributed Random Pulses", Journal of Sound & Vibration, Vol. 28, pp. 93-103, 1973.

[11] W.J. Richter Jr. and T.I. Smits, "Numerical Evaluation of Rice's Integral Representation of the Probability Density Function for Poisson Impulse Noise", J. Acoust. Soc. Am., Vol. 56, pp. 481-496, 1974.

[12] R. Lugannani and S.O. Rice, "Some Techniques for Evaluating the Amplitude Distribution of Shot Noise", Proc. Eighth Princeton Conference on Information Sciences and Systems, pp. 47-51, 1974.

[13] A. Erdélyi, Asymptotic Expansions, Dover, 1956.

[14] D. Middleton, "Man-Made Noise in Urban Environments and Transportation Systems: Models and Measurements", IEEE Trans. Comm., Vol. COM-21, pp. 1232-1241, 1973.

[15] M. Abramowitz and I. Stegun, Handbook of Mathematical Functions, U.S. Government Printing Office, Washington, 1964.

[16] S.O. Rice, "Numerical Evaluation of Integrals with Infinite Limits and Oscillating Integrands", Bell Syst. Tech. J., Vol. 54, pp. 155-164, 1975.

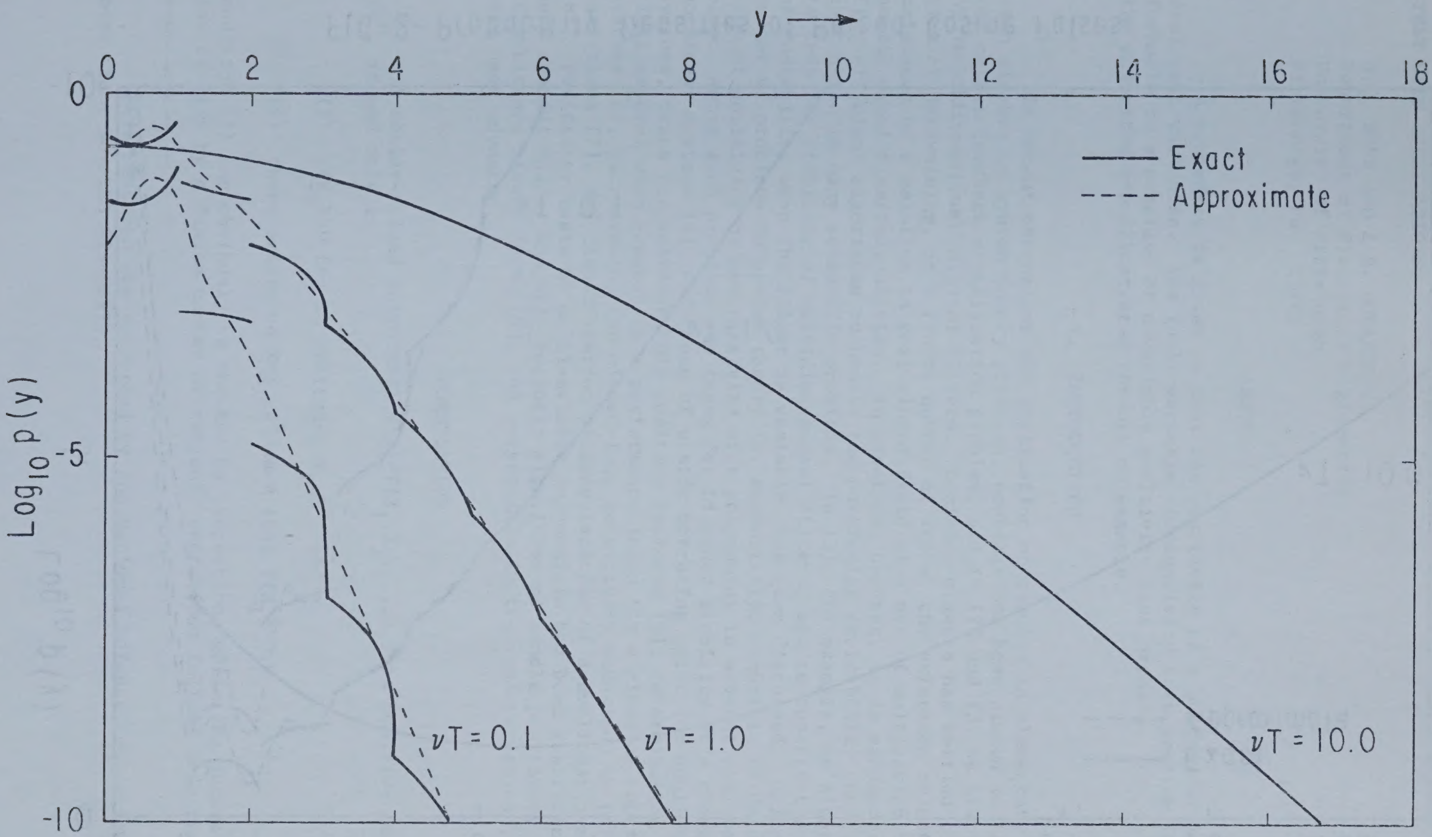


FIG. 1-Probability Densities of Sine Pulses

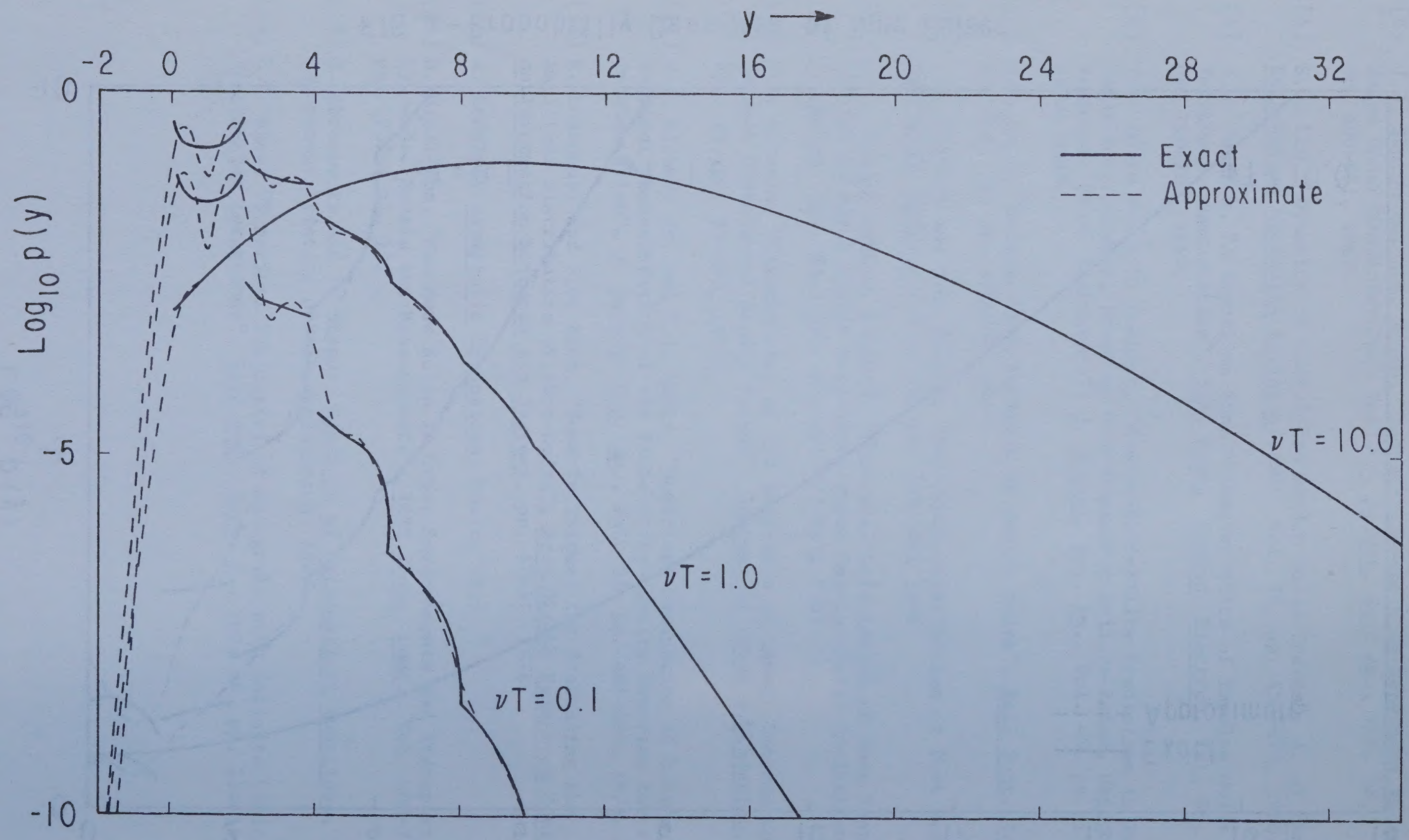


FIG. 2- Probability Densities of Raised Cosine Pulses

GENERAL PROCEDURE FOR MULTIVARIABLE POLYNOMIAL POSITIVITY AND NONNEGATIVITY TEST WITH APPLICATIONS†

N.K. BOSE AND A.R. MODARRESSI
Department of Electrical Engineering
University of Pittsburgh
Pittsburgh, Pa. 15261

ABSTRACT

A procedure is given to test for positivity of a polynomial in several real variables. The (n-1)-variable nonnegativity test problem is transformable to a subclass of n-variable positivity test problem. The details of the algorithm are illustrated through an example.

1. INTRODUCTION

In recent extensions and application of results in elementary decision algebra to system theory problems, emphasis has been placed in [1] on the output feedback stabilization problem, and in [2] and [3] on stability of multidimensional digital filters. Decision algebra has settled the question of determining, in a finite number of steps, the existence or nonexistence of a solution in real closed field of a set of multivariable polynomial equalities/inequalities. In practice, however, it is mandatory to find efficient algorithms to handle the particular decidability questions that arise in many scientific problems. In [3], for example, an algorithm to test for stability of multidimensional filters, and to construct sources of instability when the filter is unstable, has been discussed. A large number of problems in system theory is, essentially, reducible to determination of positivity or nonnegativity of a polynomial in several real variables. Among such problems are tests for Liapunov stability of a class of nonlinear systems [4], existence of static operating point for nonlinear systems, tests for multivariable positive realness [5], determination of optimal control with respect to a performance index for a class of control systems [6], guaranteeing the closed-loop sensitivity reduction to parameter variations [7], and classification of singularities of a nonlinear system [8]. Positivity tests of multivariable polynomials had been restricted to very special cases, and only recently algorithms to handle positivity tests of arbitrary binary forms [9], and polynomials in two real variables [10] have been advanced.

2. FORMULATION

A scalar-valued polynomial $V(\underline{X})=V(X_1,X_2,\ldots,X_n)$ is positive for all $\underline{X}\varepsilon R^n$, if and only if

(i) $V(\underline{X}_0)>0$ for an arbitrary $\underline{X}_0 \varepsilon R^n$, and

(ii) there exists no $\underline{X}=\underline{X}_1 \varepsilon R^n$ such that $V(\underline{X}_1)=0$.

Condition (i) can obviously be checked by inspection, while the determination of (ii) by a finite number of rational operations follows from results

---

†This research was supported by the National Science Foundation under Grant No. ENG G75-04979.

of elementary decision algebra [1]. In this paper, various aspects of an algorithm to determine whether or not condition (ii) is satisfied for any given polynomial will be investigated.

After writing $V(\underline{X})$ in recursive canonical form as a polynomial in the main variable X_n with coefficients which are polynomials in the remaining variables,

$$V(\underline{X})= \sum_{i=1}^{m} a_i X_n^i \tag{1}$$

the (2m-1)x(2m-1) matrix in (2) can be formed, whose determinant is the discriminant of $V(\underline{X})$ treated as a polynomial in X_n.

$$[\Delta^1] = \left[\begin{array}{ccccccccccc}
a_m & a_{m-1} & \cdots & \cdots & \cdots & \cdots & a_0 & 0 & \cdot & \cdot & 0 \\
0 & a_m & \cdots & \cdots & \cdots & \cdots & a_1 & a_0 & \cdot & \cdot & 0 \\
0 & \cdot & \cdots & \cdots & \cdots & \cdots & a_2 & a_1 & a_0 & \cdot & 0 \\
\cdot & \cdot & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot & \cdot \\
0 & \cdot & \cdot & \cdot & a_m & a_{m-1} & a_{m-2} & \cdot & \cdot & \cdot & a_0 \\
0 & \cdot & \cdot & \cdot & 0 & \boxed{ma_m}\,\Delta_1^1 & (m-1)a_{m-1} & \cdot & \cdot & \cdot & a_1 \\
0 & \cdot & \cdot & \cdot & ma_m & (m-1)a_{m-1} & (m-2)a_{m-2} & \Delta_3^1 & \cdot & \cdot & 0 \\
\cdot & \cdot & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot & \cdot \\
\cdot & \cdot & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot & \cdot \\
0 & ma_m & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot & 0 \\
ma_m & (m-1)a_{m-1} & \cdots & \cdots & \cdots & a_1 & 0 & \cdot & \cdot & \cdot & 0
\end{array}\right]_{\Delta^1_{2m-1}} \tag{2}$$

The result contained in the following assertion is well known.

Assertion [11, p. 48]

The number of distinct real roots, N, of a single variable polynomial,

$$F(X) = a_m X^m + a_{m-1} X^{m-1} + \dots + a_1 X + a_0 = 0, \ a_m > 0$$

with a_i's defined over a real number field, is given by

$$N=\text{var}[1,-|\Delta_1^1|,|\Delta_3^1|,\ldots,(-1)^m|\Delta_{2m-1}^1|] - \text{var}\ [1,|\Delta_1^1|,|\Delta_3^1|,\ldots,|\Delta_{2m-1}^1|] \tag{3}$$

where "var" denotes "the number of sign variations," and $|\Delta_i^1|$, i=1,2,..., 2m-1, are the signs of the "inner" determinants, Δ_i^1, in the matrix $[\Delta^1]$ in (2). Here it is assumed that $\Delta_{2m-1}^1 \neq 0$, while other critical cases occurring when $\Delta_j^1 = 0$, for j = 3,5,..,(2m-3), can be routinely taken care of [12, pp. 198-201].

In (1), when X_j, j = 1,2,...,n-1, are assigned arbitrary real values, the assertion above becomes applicable. The essence of the positivity algorithm to be discussed is to ascertain whether or not N in (3) is zero for all $(X_1,X_2,\ldots,X_{n-1})\varepsilon R^{n-1}$, where R^{n-1} is the (n-1)-dimensional real space.

3. TWO-VARIABLE POLYNOMIAL POSITIVITY

In order to explain the concepts behind the n-variable positivity test algorithm, the case of two-variable polynomials, based on the result recently given in [10], will be reviewed first.

Given $V(X_1,X_2)$, consider its recursive canonical form as a polynomial in X_2 with coefficients in terms of X_1. Assume that the highest degree in X_2 in $V(X_1,X_2)$ is m. The inner determinants Δ^1_{2k-1}, k=1,2,...,m, can be computed after the matrix $[\Delta^1]$ in (2) is formed. These inner determinants will be single variable polynomials in X_1 whose real roots can be ordered on the real line by rational operations [13]. It follows that for values of X_1 between each pair of successive roots, the signs of all the inner determinants remain invariant and can be easily determined (e.g., from an approximate plot of the inner determinants versus X_1). The assertion given above can then be used to find the number of real roots in X_2 of $V(X_1,X_2)$ when X_1 is restricted to lie in a particular open interval determined by successive ordered roots in X_1 of all the inner determinants. The singular cases, when one or more of the inner determinants is zero, have to be taken care of separately. After the completion of this process, if and only if it turns out that for all real values of X_1, N=0 and $V(0,0)>0$, then $V(X_1,X_2)>0$ for all $(X_1,X_2)\varepsilon R^2$.

4. THREE-VARIABLE POLYNOMIAL POSITIVITY

Given $V(X_1,X_2,X_3)$, consider its recursive canonical form as a polynomial in X_3 with coefficients which are two-variable polynomials in X_1 and X_2. The first-order inner determinants can be computed after $[\Delta^1]$ in (2) is formed. These will be two-variable polynomials in X_1 and X_2, and each can be written in recursive canonical form as polynomials in X_2 with coefficients in X_1. The second-order inner determinants corresponding to each first-order inner determinant can similarly be calculated. The real roots of the second order inner determinants as well as those of the resultants of pairs of the first order inner determinants (excluding the terminal ones, Δ^1_1 and Δ^1_{2m-1}) are ordered on the real line. It has been shown [16] that for an arbitrary value of X_1 in any one of the mutually disjoint open intervals thus formed on the real line, the ordering of the real roots in X_2 of the first-order inner determinants remains invariant. After the foregoing ordering in X_1 is completed, then each X_1 interval is considered with all X_2 intervals that arise when an arbitrary $X_1=X_{10}$ in that open interval is substituted in the first-order inner determinants. For any real values of X_1

and X_2 in each such "joint interval" formed, the value of N in (3) can be calculated. The singular cases, corresponding to the end points of the open intervals have to be considered separately. When N=0 for all $(X_1,X_2)\varepsilon R^2$ and $V(0,0,0)>0$, then it follows that $V(X_1,X_2,X_3)>0$ for all $(X_1,X_2,X_3)\varepsilon R^3$. The following example serves to illustrate some of the above ideas.

Example

It is required to determine whether or not the following three-variable polynomial is positive for all $(X_1,X_2,X_3)\varepsilon R^3$.

$$V(X_1,X_2,X_3) = X_1^2X_2^2 - X_3^2 - 2X_1X_2X_3^2 + X_3^4 + 2\,X_1X_2 + 2$$

Writing V in recursive canonical form as in (1),

$$V(X_1,X_2,X_3) = X_3^4 - (2X_1X_2+1)X_3^2 + (X_1^2X_2^2 + 2X_1X_2 + 2\,)$$

After $[\Delta^1]$ in (2) is formed, the first-order inner determinants are computed *exactly* using a symbolic language program [14]:

$$\Delta_1^1 = 4 \quad ; \quad \Delta_3^1 = 8(\,2X_1X_2 + 1\,) \quad ;$$

$$\Delta_5^1 = -8(\,8X_1^2X_2^2 + 18X_1X_2 + 7\,) = -8(\,2X_1X_2 + 1)(\,4X_1X_2 + 7) \;;$$

$$\Delta_7^1 = 16(\,16\,X_1^4X_2^4 + 88\,X_1^3X_2^3 + 193\,X_1^2X_2^2 + 210\,X_1X_2 + 98\,)$$

$$= 16[\,1 + (X_1X_2+1)^2](\,4X_1X_2 + 7\,)^2$$

Now, consider the second-order inner determinants corresponding to each first-order inner determinant. Clearly $\Delta_1^1>0$, and $\Delta_7^1\geq 0$ for all $(X_1,X_2)\varepsilon R^2$. Therefore, one has to consider the second-order inner determinants for Δ_3^1 and Δ_5^1 only. Considering $\Delta_3^1/8$ and $\Delta_5^1/(-8)$ as polynomials in X_2 with coefficients in terms of X_1, it is clear that the only intervals obtained by ordering of roots in X_1 of the second-order inner determinants corresponding to $\Delta_3^1/8$ and $\Delta_5^1/(-8)$ are $I_1^1=(-\infty,\ 0)$ and $I_1^2=(0,\ +\infty)$. Furthermore, since the factor $(2X_1X_2+1)$ occurs in both Δ_3^1 and Δ_5^1, the resultant of Δ_3^1 and Δ_5^1 will be identically zero, implying that for any $X_1=X_{10}<0$ the ordering of roots in X_2 of the first-order inner determinants will remain invariant. The same is true for any $X_1=X_{11}>0$. Hence, only the following two cases have to be considered:

Case (a) $\quad X_1 = X_{10}\ \varepsilon\ I_1^1 = (-\infty,\ 0)$

In Figure 1, an approximate plot of Δ_{2i-1}^1, i=1,2,3,4, versus X_2 for the chosen $X_1=X_{10}$ is shown, giving rise to three open intervals, $I_2^{(1,1)}$, $I_2^{(1,2)}$, $I_2^{(1,3)}$. For X_2 in each of these intervals, with X_1 in $(-\infty,\ 0)$, the respective values of $N=N_{1,(1,j)}$, for j=1,2,3 are, using (3), as follows:

$N_{1,(1,1)}$ = var[+,-,+,+,+] - var[+,+,+,-,+] = 2-2 = 0

$N_{1,(1,2)}$ = var[+,-,-,-,+] - var[+,+,-,+,+] = 2-2 = 0

$N_{1,(1,3)}$ = var[+,-,-,+,+] - var[+,+,-,-,+] = 2-2 = 0

Case (b) $X_1 = X_{11} \varepsilon I_1^2 (0, +\infty)$

Figure 2 shows an approximate plot of the first-order determinants versus X_2 for an arbitrary positive $X_1=X_{11}$, giving rise to intervals $I_2^{(2,1)}$, $I_2^{(2,2)}$, $I_2^{(2,3)}$. Again applying (3) the values of $N=N_{2\ (2,j)}$ for j=1,2,3 are as follows:

$N_{2,(2,1)}$ = var[+,-,-,+,+] - var[+,+,-,-,+] = 2-2 = 0

$N_{2,(2,2)}$ = var[+,-,-,-,+] - var[+,+,-,+,+] = 2-2 = 0

$N_{2,(2,3)}$ = var[+,-,+,+,+] - var[+,+,+,-,+] = 2-2 = 0

Singular Cases. The only singular case that has to be considered in this example is when $X_1=0$. This, however, is trivially taken care of when it is noted that $V(0,X_2,X_3)=X_3^4-X_3^2+2$ is always positive.

Finally, since $V(0,0,0)=2>0$, and N=0 for all real X_1,X_2, it follows that $V(X_1,X_2,X_3)>0$ for all $(X_1,X_2,X_3)\varepsilon R^3$.

5. POLYNOMIAL NONNEGATIVITY AND N-VARIABLE POSITIVITY

The following Theorem is proved and illustrated in [15].

Theorem

A multivariable polynomial $V(\underline{X})=V(X_1,X_2,\ldots X_{n-1})$ is nonnegative for all $\underline{X}\varepsilon R^{n-1}$ if and only if:

$$V_1(X_1,X_2,\ldots,X_{n-1},X_n)=V(X_1,X_2,\ldots,X_{n-1})+X_n^2$$

is positive for all $(X_1,X_2,\ldots,X_{n-1},X_n)\varepsilon R^n$, with $X_n\neq 0$. It is then seen that the test for nonnegativity can be transformed into a test for positivity of a polynomial containing an extra variable.

The generalization of the above procedure to test for n-variable positivity is conceptually straightforward, although in practice the limitations of computer time and storage set an upper bound on the number of variables as well as the highest power of each variable. For an n-dimensional polynomial, the essence of the algorithm is to successively compute all the i^{th}-order inner determinants for i=1,2,...,n-1, order the real roots of the (n-1)-order inner determinants as well as the relevant resultants and work backward to the original polynomial, through a set of joint intervals created as a result of the ordering at each stage.

6. CONCLUSION

A procedure has been given to test a multivariable polynomial for positivity. The resolution of the various types of singularities, when one or more of the inner determinants are zero, has only been alluded to, with detailed discussions contained in a forthcoming article [16]. The nonnegativity test problem is transformable to a class of positivity test problem [15]. The details of the procedure for a three-variable polynomial have been illustrated through an example. In the example, although it is possible to consider $V(X_1,X_2,X_3)$ in recursive canonical form as a polynomial in X_2, which has the highest power of 2, it has been considered as a polynomial in X_3 to illustrate more clearly the concept of joint intervals.

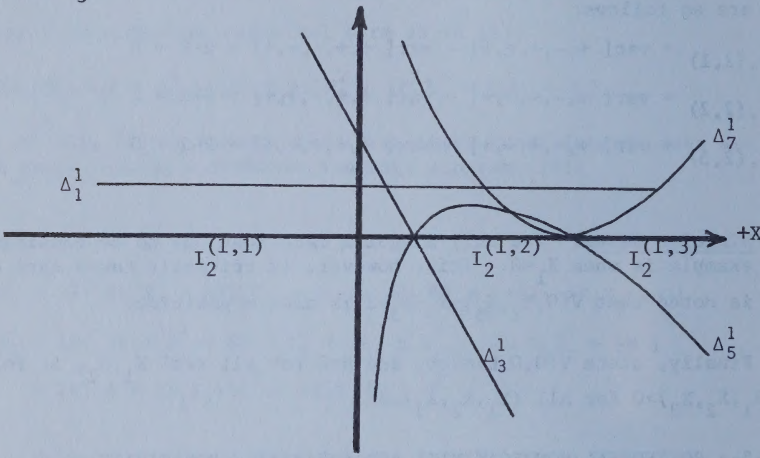

Δ_7^1
Δ_1^1
$+X_2$
$I_2^{(1,1)}$
$I_2^{(1,2)}$
$I_2^{(1,3)}$
Δ_5^1
Δ_3^1

Figure 1

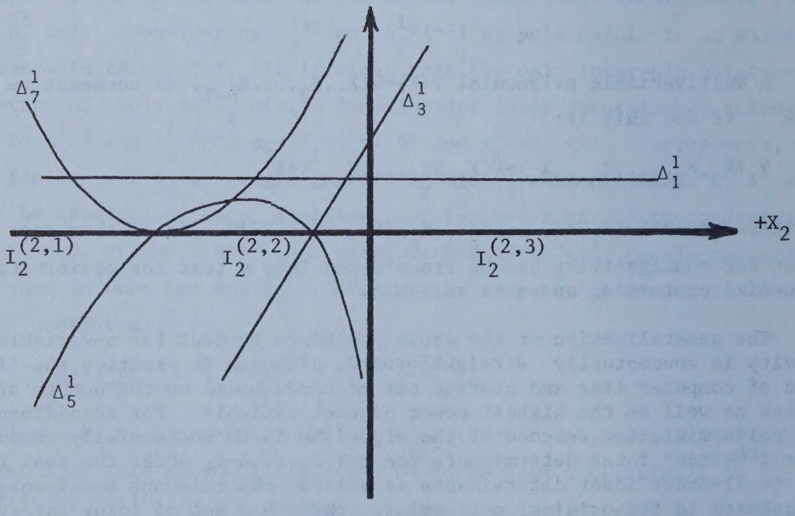

Δ_7^1
Δ_3^1
Δ_1^1
$+X_2$
$I_2^{(2,1)}$
$I_2^{(2,2)}$
$I_2^{(2,3)}$
Δ_5^1

Figure 2

REFERENCES

[1] B.D.O. Anderson, N.K. Bose and E.I. Jury, "Output Feedback Stabilization Problems - Solution Via Decision Methods," IEEE Trans. Auto. Control, Vol. AC-20, Feb., 1975, pp. 53-66.

[2] N.K. Bose and E.I. Jury, "Positivity and Stability Tests for Multidimensional Filters (Discrete-Continuous)," IEEE Trans. Acoustics, Speech, and Signal Processing, June, 1974, pp. 174-180.

[3] N.K. Bose and P.S. Kamat, "Algorithm for Stability Test of Multidimensional Digital Filters," IEEE Trans. Acoustics, Speech, and Signal Processing, Oct., 1974, pp. 307-314.

[4] J.C. Hsu and A.U. Meyer, Modern Control Principles and Applications, McGraw Hill Book Co., New York, 1968.

[5] N.K. Bose and R.W. Newcomb, "Tellegen's Theorem and Multivariable Realizability Theory," Int. Jour. Electronics, Vol. 36, Mar., 1974, pp. 417-425.

[6] B.D.O. Anderson, "The Inverse Problem of Optimal Control," Rept. SEL-66-038 (TR-6560-3), Stanford Electronics Laboratories, Stanford, CA, 1966.

[7] W.R. Perkins and J.B. Cruz, Jr., "The Parameter Variation Problem in State Feedback Control Systems," Trans. ASME, J. Basic Engrg., Vol. 87, March, 1965, pp. 25-30.

[8] A. Blaquiere, Nonlinear System Analysis, Academic Press, New York, 1966.

[9] N.K. Bose and E.I. Jury, "Inner Algorithm to Test for Positive Definiteness of Arbitrary Binary Forms," IEEE Trans. Auto. Control, Vol. AC-20, Feb., 1975, pp. 169-170.

[10] N.K. Bose, "Test for Liapunov Stability by Rational Operations," IEEE Trans. Auto. Control, Vol. AC-20, Oct., 1975.

[11] E.I. Jury, Inners and Stability of Dynamic Systems, John Wiley and Sons, New York, 1974.

[12] F.R. Gantmacher, The Theory of Matrices, Vol. 2, Chelsea Publications Co., New York, 1964.

[13] B.E. Meserve, "Inequalities of Higher Degrees in One Unknown," Amer. J. Math., Vol. 49, pp. 357-370, 1947.

[14] A.C. Hearn, "Reduce 2: A System and Language of Algebraic Manipulations," ACM Proc. 2nd. Symp. Symbolic and Algebraic Manipulations, Los Angeles, March, 1971, pp. 128-133.

[15] A.R. Modarressi and N.K. Bose, "A Multivariable Polynomial Nonnegativity Test," Proc. IEEE, Dec., 1975.

[16] N.K. Bose and A.R. Modarressi, "General Procedure for Multivariable Polynomial Positivity Test with Control Applications," to appear.

PULSE TO IMPULSE COMPRESSION VIA DISTRIBUTED FEEDBACK*

D. HAZONY
Department of Electrical Engineering and Applied Physics
Case Western Reserve University
Cleveland, Ohio 44106

ABSTRACT

A pulse to impulse compressor is the ideal tool for separating overlapping prescribed time limited signals. A synthesis procedure is provided for obtaining distributed networks capable of compressing any prescribed symmetric time limited signal into an impulse. The method relies on a tapered feedback defined by the shape of the signal.

1. Introduction

Distributed feedback can be used to provide an ideal pulse to impulse compression in a direct manner. The concept is illustrated in Fig. 1 where the emitter follower implies a unity gain amplifier. The resistive path provides a sufficiently low impedance so that essentially the voltage at each leg of the ladder is given by $E(x) = \alpha(x)E_2$ where

$$\alpha(x) = \int_0^x \rho(x)\,dx \Big/ \int_0^\lambda \rho(x)\,dx \tag{1}$$

A possible realization is shown in Fig. 2.

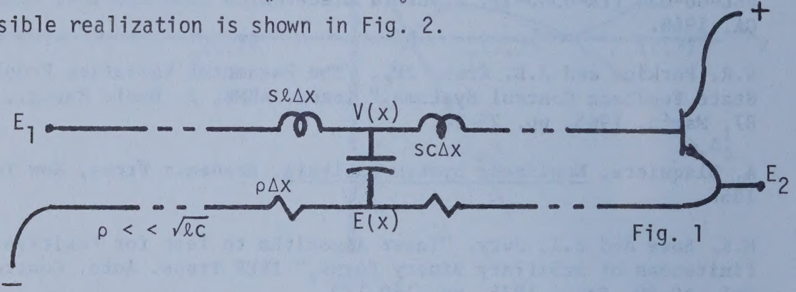


Fig. 1

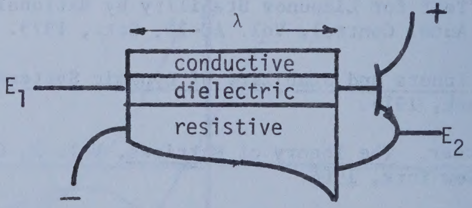


Fig. 2

Without a loss in generality let $\int_0^\lambda \rho(x)\,dx = 1$ and let $\rho(x)$ define the taper of this network.

A similar arrangement was proposed in Reference 1 to realize the voltage transfer function $E_2 = E_1 e^{-s\tau}$ by an RC network.

2. Development

Considering the network of Fig. 1 one may write

---

*Work partially supported by National Science Foundation Grant GK42720.

$$\frac{dV}{dx} + \ell s\ I = 0 \tag{2}$$

$$\frac{dI}{dx} + sc[V - \alpha(x)\ E_2(s)] = 0 \tag{3}$$

Differentiating in respect to x permits the elimination of the current I above to give:

$$\frac{d^2V}{dx^2} - s^2\ \tau^2 V = -\ s^2\ \tau^2\ \alpha(x)\ E_2(s)\ ,\quad \tau = \sqrt{\ell c} \tag{4}$$

We proceed to write the general solution of this equation in terms of two arbitrary constants C_1 and C_2 :

$$V(x,s) = -\ s\tau\ E_2(s) \int_0^x \alpha(t) \sinh(x-t)\ s\tau\ dt$$

$$+\ C_1 \cos hs\ \tau x + C_2 \sin hs\ \tau x \tag{5}$$

where

$$V(o,s) = E_1(s)\ , \tag{6a}$$

$$I(\lambda,s) = I_{out} = -\ \frac{1}{\ell s} \left.\frac{dV}{dx}\right|_{x=\lambda} = 0 \tag{6b}$$

$$V(\lambda,s) = E_2(s)\ . \tag{6c}$$

Equation (6) can be used to solve for C_1 and C_2 . The following results:

$$t_{21} = \frac{E_2(s)}{E_1(s)} = [\cos hs\ \tau\lambda - s\tau \int_0^\lambda \alpha(x) \sin hs\ \tau x\ dx]^{-1} \tag{7}$$

Letting $\alpha(0)$, $\alpha(\lambda) = 1$, and $\rho = d\alpha/dx$ permits rewriting (7) as:

$$t_{21} = [\int_0^\lambda \rho(x) \cosh s\tau x\ dx]^{-1} \tag{8}$$

3. The Pulse to Impulse Compressor

Consider a time limited symmetric pulse define by:

$$f(t) = 0\ ,\quad 0 > t > 2T \tag{9a}$$

$$f(T+x) = f(T-x)\ ,\quad 0 \le x \le T \tag{9b}$$

The Laplace transform of this signal is

$$F(s) = \int_0^{2T} f(t)\ e^{-st}\ dt$$

$$= \int_{-T}^{T} f(T+x)\ e^{-s(x+T)}\ dx \tag{10}$$

Taking advantage of (9b) gives

$$F(s) = e^{-sT}\ 2 \int_0^T f(T+x) \cosh sx\ dx$$

$$\equiv e^{-sT}\ F_1(s)$$

Suppose now that the signal is applied to the filter above. It would reduce into an impulse (centered at T) if $F_1(s)$ is cancelled. Hence, the following must be valid.

$$\int_0^\lambda \rho(x) \cosh s\tau x\ dx = 2 \int_0^T f(T+x) \cosh sx\ dx \tag{11}$$

A variable change $x \to \tau x$ gives $\lambda = T/\tau$ and

$$\rho(x/\tau) = 2\tau f(T+x)$$

This completes the derivation.

4. Discussion

Note that the taper $\rho(x)$ is obtained directly from the shape of the pulse. A similar concept was used by F. Bolinder as early as 1950 (2) to synthesize transmission line tapers with a specified reflection coefficient.

REFERENCES

1. D. Hazony, and R. Sobecks, "A distributed RC active delay line," SIAM J. Appl. Math. Vol. 22, No. 2, March 1972, pp. 139-141.

2. E. F. Bolinder, "Fourier transform in the theory of inhomogeneous transmission lines," Proc. IRE, Vol. 38, Nov. 1950, p. 1354.

ON THE ANALYSIS AND SYNTHESIS OF A CLASS OF NONLINEAR CIRCUITS

M. LAL and K. SAM SHANMUGAM
Department of Electrical Engineering
Wichita State University, Wichita, Kansas 67208

ABSTRACT

The problem of analysis and synthesis of a class of systems, and networks containing single amnesic type nonlinearity is discussed. A synthesis approach for such a class, based on Volterra series representation of the input/output behavior of nonlinear systems, is given. The realizability conditions and a synthesis procedure are developed for realizing the specified linear and nonlinear transfer functions or kernels of the Volterra series in frequency domain.

I. INTRODUCTION

An interesting approach, which has received considerable attention for the analysis of nonlinear circuits and systems, is based on the Volterra series representation of the input/output behavior of a nonlinear system [1]. The original application of Volterra functionals to the analysis of nonlinear circuits, proposed by Wiener [2], is particularly suited for mild excitations of a fairly general nature. This approach has been successfully used for the analysis of distortion and experimental cross-modulation characterization of amplifiers [3-5]. Bedrosian and Rice [6] applied this approach to systems driven by sine waves and Gaussian noise and recently Bussgang et al [7] applied it to the analysis of nonlinear systems with multiple inputs. The frequency-domain representation of the Volterra series has also been used for the synthesis of a class of nonlinear systems [8] and for the identification of the structure of nonlinear blocks in communication systems [9].

In the present paper, an attempt is made to apply the Volterra series approach for the synthesis of a class of nonlinear systems and circuits. In particular, given a set of nonlinear transfer functions [7], a set of realizability conditions and a realization procedure are developed for a simple class of nonlinear systems and for circuits containing a single amnesic, power series type nonlinearity. The synthesis procedure is illustrated by means of examples.

II. FREQUENCY DOMAIN REPRESENTATION OF NONLINEAR SYSTEMS

Wiener suggested the following Volterra series representation for the output y(t) of a nonlinear system in terms of the input x(t).

$$y(t) = \sum_{k=1}^{\infty} y_k(t) \quad , \tag{1}$$

where,

$$y_k(t) = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} h^{(k)}(\tau_1,\tau_2,\ldots,\tau_k) \prod_{i=1}^{k} x(t-\tau_i)d\tau_i . \tag{2}$$

The $h^{(k)}(\tau_1,\tau_2,\ldots,\tau_k)$ in (2) is called the k^{th} order kernel (impulse response). This representation can be viewed as a generalization of the convolution representation of linear systems.

In the frequency domain, the system can be represented by [7]

$$Y(f) = \sum_{k=1}^{\infty} Y_k(f) \quad ,$$

where

$$Y_k(f) = \int_{-\infty}^{\infty}\!\!\ldots\!\int H^{(k)}(f_1,\ldots,f_k)\,\delta(f-f_1-f_2-\ldots-f_k) \prod_{i=1}^{k} X(f_i)df_i . \tag{3}$$

In (3), $H^{(k)}(f_1,f_2,\ldots,f_k)$ is the Fourier transform of the k^{th} order kernel $h^{(k)}(\tau_1,\tau_2,\ldots,\tau_k)$, and is called the nonlinear transfer function of order k. For a given system or circuit, the nonlinear transfer functions can either be determined analytically or experimentally by applying a multitone signal at the input to the system and measuring the response at certain frequencies [7,9].

III. SYNTHESIS OF A CLASS OF NONLINEAR SYSTEMS

In this section, we present certain results concerning the realizability of a class of nonlinear systems.

If a given set of transfer functions $H^{(k)}(f_1,f_2,\ldots,f_k)$ admit the separation:

$$H^{(k)}(f_1,f_2,\ldots,f_k) = C_k H_k(f_1) H_k(f_2) \ldots H_k(f_k) G_k(f_1+f_2+\ldots+f_k) \tag{4}$$

$$(k=1,2,3,\ldots),$$

a nonlinear system for the specified transfer functions can be realized as shown in Fig. 1.

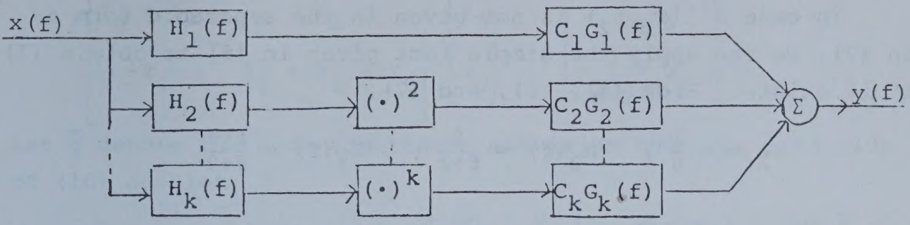


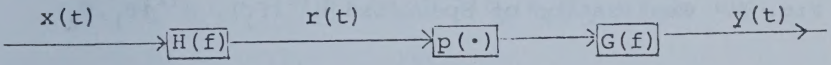
Fig. 1. A Nonlinear System Realization.

This result can be proved by substituting for $H^{(k)}$ from (4) in (3) and simplifying the resulting expression [8].

If $H^{(k)}$ admits the separation:

$$H^{(k)}(f_1,f_2,\dots,f_k) = C_k H(f_1)H(f_2)\dots H(f_k)G(f_1+f_2+\dots+f_k), \quad (5)$$

the realization of the nonlinear system reduces to the structure shown in Fig. 2 [10].



$$p[r(t)] = \sum_{k=1}^{\infty} C_k r^k(t).$$

Fig. 2. Power Series Realization.

Several other special structures, corresponding to the cases when $H_k(f) = 1$ or $G_k(f_1 + f_2 + \dots + f_k) = 1$ in (4), can easily be obtained.

A simple procedure to test whether a specified set of nonlinear transfer functions admit the separation in (4) or (5) has been given in [8].

Example 1.

Suppose the first- and second-order transfer functions are given as [9]

$$H^{(1)}(j\omega_1) = \frac{1}{2}\,\frac{j\omega_1}{j\omega + 2}, \quad \text{or} \quad H^{(1)}(f_1) = \frac{1}{2}\,\frac{f_1}{f_1 + 2} \quad (6)$$

$$H^{(2)}(j\omega_1,j\omega_2) = -\frac{1}{8}\left(\frac{j\omega_1}{j\omega_1 + 2}\right)\left(\frac{j\omega_2}{j\omega_2 + 2}\right)\left(\frac{j\omega_1 + j\omega_2 + 4}{j\omega_1 + j\omega_2 + 2}\right)$$

$$\text{or,}\ H^{(2)}(f_1,f_2) = -\frac{1}{8}\left(\frac{f_1}{f_1 + 2}\right)\left(\frac{f_2}{f_2 + 2}\right)\left(\frac{f_1 + f_2 + 4}{f_1 + f_2 + 2}\right) \quad (7)$$

In case $H^{(2)}(f_1,f_2)$ is not given in the separable form as in (7), we can apply the simple test given in [8] to obtain (7) if it exists. From (4), (6), and (7),

$$C_2 = -\frac{1}{8}, \quad H_2(f) = \frac{f}{f+2}, \quad G_2(f) = \frac{f+4}{f+2}$$

$$C_1 = \frac{1}{2}, \quad H_1(f) = \frac{f}{f+2}, \quad G_1(f) = 1 \tag{8}$$

Thus, a nonlinear system corresponding to the specified transfer functions can be realized as shown in Fig. 3.

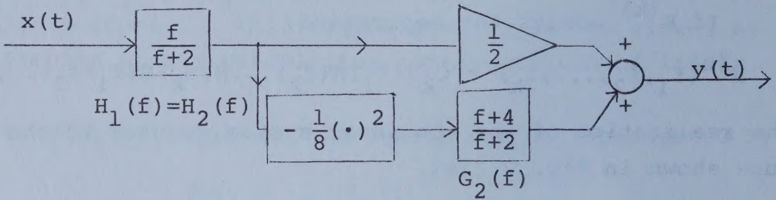


Fig. 3. Realization of Specified $H^{(1)}(f_1)$, $H^{(2)}(f_1,f_2)$.

IV. SYNTHESIS OF A CLASS OF NONLINEAR CIRCUITS

In this section, we attempt to synthesize simple nonlinear L-section for specified source and load admittance, and the transfer functions $H^{(1)}(f_1)$, $H^{(2)}(f_1,f_2)$, etc. This circuit is shown in Fig. 4 and it is assumed that the nonlinear element has amnesic, power-series type nonlinearity (nonlinear resistor) described by

$$i_R(t) = K[v_0(t)] = K_1 v_0(t) + K_2 v_0^2(t) + \ldots \tag{9}$$

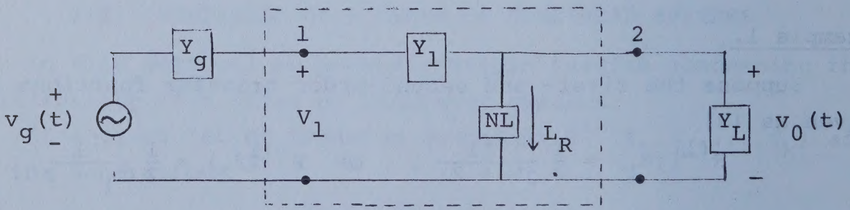


Fig. 4. L-Section Containing a Nonlinear Element.

The node equations in the frequency domain at nodes 1 and 2 can be written as

$$\begin{bmatrix} Y_g + Y_1 & -Y_1 \\ -Y_1 & K_1 + Y_1 + Y_L \end{bmatrix} \begin{bmatrix} V_1 \\ V_0 \end{bmatrix} = \begin{bmatrix} Y_g V_g \\ -\sum_{n=2}^{\infty} K_n V_0^{\ n} \end{bmatrix} \tag{10}$$

Let $\overline{Y}$ denote the node-admittance matrix on the left hand side of (10) and let

$$\overline{Y}^{-1} = \begin{bmatrix} z_1 & z_2 \\ z_3 & z_4 \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} K_1+Y_1+Y_L & Y_1 \\ Y_1 & Y_g+Y_1 \end{bmatrix}, \quad \text{where } \Delta = \det \overline{Y} = K_1(Y_g+Y_1) + Y_g Y_1 + Y_L Y_1 + Y_g Y_L \tag{11}$$

Assuming $v_g(t) = e^{j2\pi ft}$, we can obtain the linear transfer function, $H^{(1)}(f_1) = \frac{V_0(f_1)}{V_g(f_1)}$, and $H^{(2)}(f_1,f_2)$, etc., from the analysis based on Volterra series representation [7], as

$$H^{(1)}(f_1) = z_2(f_1) Y_g(f_1) \tag{12a}$$

$$H^{(2)}(f_1,f_2) = -K_2 H^{(1)}(f_1) H^{(1)}(f_2) z_4(f_1 + f_2), \text{ etc.} \tag{12b}$$

Conversely, given $H^{(1)}(f_1)$, $H^{(2)}(f_1,f_2)$, etc., Y_g, Y_L, we can easily verify if separation as in (12) is possible. Moreover, in case such a separation is possible, we may be able to realize the circuit shown in Fig. 4 for the given specifications, as discussed next.

Given $H^{(1)}(f_1)$, $H^{(2)}(f_1,f_2)$, Y_g, Y_L, we can easily verify if the separation as in (12) is possible. If so, we can obtain

$$G(f_1 + f_2) \triangleq \frac{H^{(2)}(f_1,f_2)}{H^{(1)}(f_1) H^{(1)}(f_2)} . \tag{13}$$

From (12a), (12b), and (11), we have

$$Y_1(f) = \frac{-K_2 Y_g H^{(1)}(f)}{Y_g G(f) + K_2 H^{(1)}(f)} \tag{14}$$

which should be positive real function if $Y_1(f)$ is to be realized by passive elements.

Next, substituting for $Y_1(f)$ from (14) and $z_2(f)$ from (11) in (12a) and simplifying, we have

$$K_1 = K_2 \left(\frac{H^{(1)}(f) - 1}{G(f)}\right) - Y_L \tag{15}$$

In (15) if we can choose a constant K_2 such that the right hand side of (15) is constant, and if $Y_1(f)$ in (14) is p.r., the given specifications can be realized by the terminated L-section shown in Fig. 4.

As an example, consider the $H^{(1)}(f)$ and $H^{(2)}(f_1,f_2)$ specified as in (6) and (7) in example 1. Let us assume the source and load admittances Y_g, Y_f to be unity. G(f) in this case can be obtained as

$$G(f) = -\frac{1}{2}\frac{f+4}{f+2}\ . \tag{16}$$

Substituting for $H^{(1)}(f)$, G(f) and Y_L in (15),

$$K_1 = K_2 - 1 \tag{17}$$

From (14),

$$Y_1(f) = \frac{K_2 f}{(1 - K_2)f + 4}\ . \tag{18}$$

$Y_1(f)$ in (18) is p.r. for $(1 - K_2) \geq 0$. Different realizations can be obtained corresponding to different values of K_2 satisfying this constraint. For example, for

$$K_2 = 1\ , \quad K_1 = 0\ , \quad Y_1(f) = \frac{f}{4}\ , \tag{19a}$$

$$K_2 = \frac{1}{2}\ , \quad K_1 = -\frac{1}{2}\ , \quad Y_1(f) = \frac{f}{f+8}\ ; \tag{19b}$$

the two realizations corresponding to the values in (19a) and (19b) are shown in Fig. 5(a) and 5(b).

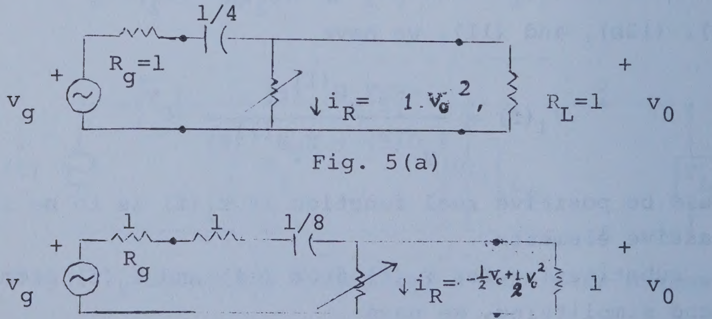

1/4
$R_g=1$
+
v_g
$\downarrow i_R = 1 \cdot v_c^2$,
$R_L=1$
+
v_0
1
1
1/8
+
R_g
v_g
$\downarrow i_R = -\frac{1}{2}v_c + \frac{1}{2}v_c^2$
1
+
v_0

Fig. 5(a)

Fig. 5(b)

Fig. 5. Two Network Realizations.

V. CONCLUSIONS

A synthesis approach for a class of nonlinear systems, based on Volterra series representation, has been presented. An attempt has been made to use this approach for the synthesis of simple nonlinear circuits. Although the realization of simple L-section containing a nonlinear resistor and terminated at both the ports was discussed in the initial attempt, it seems that the approach can be used for the synthesis of different network structures containing other types of nonlinearities. It is hoped that further work in this direction might lead to more fruitful results.

REFERENCES

1. V. Volterra, Theory of Functionals and of Integral and Integro-Differential Equations. New York: Dover, 1959,

2. N. Wiener, "Response of a nonlinear device to noise," M.I.T. Radiation Lab., Rep. V-165, April 1942.

3. S. Narayanan, "Application of Volterra series to intermodulation distortion analysis of transistor feedback amplifiers," IEEE Trans. Circuit Theory, vol. CT-17, pp. 518-527, Nov. 1970.

4. S. Narayanan and H. C. Poon, "An analysis of distortion in bipolar transistors using integral charge control model and Volterra series," IEEE Trans. Circuit Theory, vol. CT-20, pp. 212-219, July 1973.

5. R. Meyer, M. J. Shensa, and R. Eschenback, "Cross-modulation and intermodulation in amplifiers at high frequencies," IEEE J. Solid State Circuits, vol. SC-7, pp. 16-23, Feb. 1972.

6. E. Bedrosian and S. O. Rice, "The output properties of Volterra Systems (nonlinear systems with memory) driven by harmonics and Gaussian inputs," Proc. IEEE, vol. 59, pp. 1688-1707, Dec. 1971.

7. J. J. Bussgang, L. Ehrman, and J. W. Graham, "Analysis of nonlinear systems with multiple inputs," Proc. IEEE, vol. 62, pp. 1088-1119, Aug. 1974.

8. K. Sam Shanmugam and M. Lal, "Analysis and Synthesis of a Class of nonlinear systems," to appear in IEEE Trans. Circuits and Systems, Jan. 1976.

9. K. Sam Shanmugam, Mark T. Jong, and M. Lal, "On the identification of the structure of nonlinear blocks in communication systems," to be presented at the 1975 N.E.C., Chicago, Oct. 1975.

10. A. M. Bush, "Some Aspects of the Synthesis of Nonlinear Systems," M.I.T. Technical Report No. 441, March 1966.

A SEQUENTIAL METHOD OF SOURCE TRANSFORMATIONS

SHOJI SHINODA
Faculty of Science and Engineering, Chuo University, Tokyo
TSUYOSHI MARUYAMA
Planning Bureau, Science and Technology Agency, Tokyo
and
YOJI KAJITANI
Faculty of Engineering, Tokyo Institute of Technology, Tokyo

ABSTRACT

A new method of source transformations on a linear LCR network, which will be called a sequential method of source transformations, is defined and formulated.

INTRODUCTION

As is seen in many books on network analysis [1], source transformations are one of the fundamental concepts in network theory. It is often desired to seek a special distribution of sources under source transformations.

In this paper, a new method of source transformation, which allow us to relocate sources in a linear LCR network without changing all the voltages and currents of immittance elements of the network, is defined and formulated. The method is called a sequential method of source transformations, and is worth notice in close relation to the formulation of the sequential method of network analysis.

PRELIMINARIES

Throughout this paper, we consider a linear LCR network N whose topological structure is a nonseparable oriented graph G with m + 1 nodes and n branches, and whose source distribution satisfies the following conditions:

(a) each voltage source is connected in series with an immittance element if all current sources are open circuited.

(b) each current source is connected in parallel with an immittance element if all voltage sources are short circuited.

Without loss of generality, it is assumed that a typical branch, say branch a (a = 1, ---, n), of N is of the form shown in (a) or (b) of Fig. 1, where the rectangular box represents an immittance element and

e_a: the voltage of the voltage source of branch a,

c_a: the current of the current source of branch a,

v_a: the voltage of the immittance element of branch a,

i_a: the current of the immittance element of branch a,

$\overline{V}_a = \nu_a - e_a$: the terminal voltage of branch a,

$I_a = i_a - c_a$: the terminal current of branch a,

y_{aa}: the admittance of the immittance element of branch a,

Z_{aa}: the impedance of the immittance element of branch a, and $y_{aa}Z_{aa} = 1$.

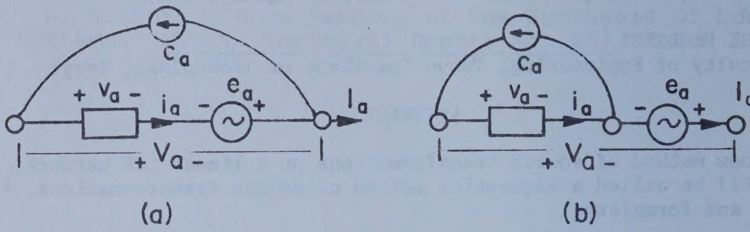


Fig. 1. Branch a of N, including voltage anc current sources.

The column-vectors whose a-th components are e_a, c_a, v_a, i_a, V_a and I_a are represented by $e = [e_1, \ldots, e_n]^t$, $c = [c_1, \ldots, c_n]^t$, $v = v_1, \ldots, v_n]^t$, $i = [i_1, \ldots, i_n]^t$, $V = [V_1, \ldots, V_n]^t$, and $I = [I_1, \ldots, I_n]^t$, respectively, where the superscript "t" is used to denote the transposition as a matrix, and these column-vectors are called voltage source vector, current source vector, element voltage vector, element current vector, branch voltage vector, and branch current vector of N, respectively. The diagonal matrices whose a-th diagonal elements are y_{aa} and $_{aa}$ are denoted by $y = \mathrm{diag}[y_{11}, y_{22}, \ldots, y_{nn}]$ and $2 = \mathrm{diag}[Z_{11}, Z_{22}, \ldots, Z_{nn}]$, respectively, and they are called the branch admittance matrix and the impedance matrix of N, respectively. Here, there holds $Zy = yZ = 1$ stands for a unit matrix.

Since G is a connected graph with m + 1 nodes and n branches, the number of maximally linearly independent cutsets of G is m and the number of maximally linearly independent tiesets of G is k = n - m. The symbol Q is used to denote a cutset matrix of G which gives all the information about the connectivity relations of the m linearly indepedent outsets and the n branches of G, and the symbol R is used to denote a tieset matrix of G which gives all the information about the connectivity relations of the k linearly independent tiesets and the n branches of G. Then, as one of the most fundamental properties of G,

$$Q R^t = 0 \quad \text{and/or} \quad R Q^t = 0 \tag{1}$$

are satisfied where 0 stands for the zero matrix.

With the above descriptions, Kirchhott's current law (KCL) and voltage (KVL) can be expressed as

$$Qi = Qc \tag{2}$$

and

$$Rv = Re \ , \tag{3}$$

respectively, and Ohm's law (OL) can be expressed as

$$Li = Mv \tag{4}$$

where L and M are appropriate diagonal matrices such that $L = M\mathbf{Z}$, $M = Ly$, $\det(M) \neq 0$ and $\det(L) \neq 0$ where $\det(A)$ is used to denote the determinant of a sequare matrix A.

Now, let us consider the relations

$$\hat{c} = c + R^t J^{(0)} \tag{5}$$

$$\hat{e} = e + Q^t U^{(0)} \tag{6}$$

where $J^{(0)} = [J_1^{(0)}, \ldots, J_k^{(0)}]^t$ and $U^{(0)} = [U_1^{(0)}, \ldots, U_m^{(0)}]^t$ are column-vectors whose components may be assigned arbitrary values. Then, substituting (5) into (2), we have

$$\begin{aligned} Qi &= Qc \\ &= Q(\hat{c} - R^t J^{(0)}) \\ &= Q\hat{c} - Q R^t J^{(0)} \\ &= Q\hat{c} \end{aligned} \tag{7}$$

and

$$\begin{aligned} Rv &= Re \\ &= R(\hat{e} - Q^t U^{(0)}) \\ &= R\hat{e} - R Q^t U^{(0)} \\ &= R\hat{e} \end{aligned} \tag{8}$$

Thus we see that i and v remains unchanged under the transformations of c and e into $\hat{c}$ and $\hat{e}$. (5) and (6) are called the guage transformations of current and voltage sources, respectively. $\hat{c}$ and $\hat{e}$ are new current source vector and voltage source vector obtained from c and e after the guage transformations, respectively. $\hat{N}$ is used to denote the network obtained from N by the guage transformations, and a typical branch, say branch a, of $\hat{N}$ is of the form shown in (a) or (b) of Fig. 2, where $\hat{c}_a$ is the a-th component of $\hat{c}$, and $\hat{e}_a$ is the a-th component of $\hat{e}$.

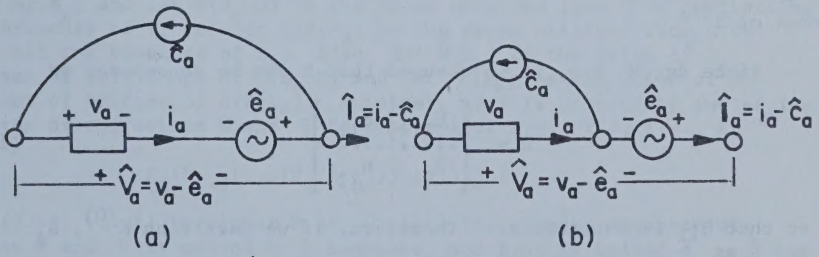

$\hat{c}_a$
$+\ v_a\ -$
i_a
$-\ \hat{e}_a\ +$
$\hat{i}_a = i_a - \hat{c}_a$
$+\ \hat{V}_a = v_a - \hat{e}_a\ -$
(a)
$\hat{c}_a$
v_a
i_a
$-\ \hat{e}_a\ +$
$\hat{i}_a = i_a - \hat{c}_a$
$+\ \hat{V}_a = v_a - \hat{e}_a\ -$
(b)

Fig. 2. Branch a of $\hat{N}$, including voltage and current sources

FORMULATION OF A SEQUENTIAL METHOD OF SOURCE TRANSFORMATIONS

KCL equation of N can be written from (7) as

$$i = \hat{c} + R^t J^{(1)} \tag{9}$$

where $J^{(1)} = [J_1^{(1)}, \ldots, J_k^{(1)}]^t$ is a column-vector whose components called tieset currents of $\hat{N}$. Similarly, KVL equation of $\hat{N}$ can be written from (8) as

$$v = \hat{e} + Q^t U^{(1)} \tag{10}$$

where $U^{(1)} = [U_1^{(1)}, \ldots, U_m^{(1)}]^t$ is a column-vector whose components called cutset voltages of $\hat{N}$.

Substituting (9) and (10) into (4), we obtain

$$H X^{(1)} = \hat{S} \tag{11}$$

where H is a matrix obtained from the matrix $[M Q^t \vdots - L R^t]$ by suitable permutation of its columns, $X^{(1)}$ is a column-vector obtained from the vector $\begin{bmatrix} U^{(1)} \\ \cdots\cdots \\ J^{(1)} \end{bmatrix}$ by the corresponding permutation of its rows, and

$$\hat{S} = L\hat{c} - M\hat{e} \ . \tag{12}$$

Here, (11) is called a description equation of $\hat{N}$ and $\hat{S}$ is called an excitation vector of $\hat{N}$. If $L = 1$ and $M = y$, then $\hat{S}$ is the column-vector whose components are called the equivalent current sources of $\hat{N}$. If $L = \mathbf{Z}$ and $M = 1$, then $\hat{S}$ is the column-vector whose components are called the equivalent voltage sources of $\hat{N}$.

In network analysis, it is often desired to seek the column-vector $\hat{S}$ that contains as many zeros as possible under some conditions.

Now, substituting (4) and (5) into (12), we obtain

$$\hat{S} = S - H X^{(0)} \tag{13}$$

where $X^{(0)}$ is a column-vector which is obtained from the vector $\begin{bmatrix} U^{(0)} \\ \cdots\cdots \\ J^{(0)} \end{bmatrix}$ and whose rows are arranged in the same order as the rows of $X^{(1)}$.

Since $\det(H) \neq 0$, it is assumed that H can be decomposed as

$$H = \left[\begin{array}{c:c} H_{11} & H_{12} \\ \hdashline H_{21} & H_{22} \end{array}\right] \tag{14}$$

so that H_{11} is nonsingular. Therefore, if we partition $X^{(0)}$, S, and $\hat{S}$ as

$$X^{(0)} = \begin{bmatrix} X_1^{(0)} \\ \cdots \\ X_2^{(0)} \end{bmatrix} , \quad S = \begin{bmatrix} S_1 \\ \cdots \\ S_2 \end{bmatrix} , \text{ and } \quad S = \begin{bmatrix} \hat{S}_1 \\ \cdots \\ \hat{S}_2 \end{bmatrix} \tag{15}$$

corresponding to the decomposition of H, then (13) can be rewritten as

$$\hat{S}_1 = S_1 - H_{11}X_1^{(0)} - H_{12}X_2^{(0)} \tag{16}$$

$$\hat{S}_2 = S_2 - H_{21}X_1^{(0)} - H_{22}X_2^{(0)} \tag{17}$$

Here, if we set

$$H_{11}X_1^{(0)} = S_1 \tag{18}$$

$$X_2^{(0)} = 0 \tag{19}$$

then we obtain

$$\hat{S}_1 = 0 \tag{20}$$

$$\hat{S}_2 = S_2 - H_{21}H_{11}^{-1}S_1 \tag{21}$$

where H_{11}^{-1} is the inverse of H_{11} and 0 stands for a zero vector. Moreover, if H_{11} is a triangular matrix, then one of the components of $X_1^{(0)}$ can be determined by solving one equation in one unknown, and the remaining components of $X_1^{(0)}$ can be sequentially found only by substitution. Thus, it seems to be adequate as a preprocessing procedure of network analysis to express $\hat{S}$ as (20) and (21) under the condition that H_{11} is a triangular matrix. Hereafter, the above method of guage transformations under the condition that H_{11} is a triangular matrix is called a sequential method of source transformations and is considered from the graph-theoretic point of view.

Let us denote the set of branches of G by B, let us denote the set of branches corresponding to the rows of H_{11} by $B(H_{11})$ and let us denote the set of branches corresponding to the rows of H_{21} by $B(H_{21})$. Then, $B = B(H_{11}) \cup B(H_{21})$ and $B(H_{11}) \cap B(H_{21}) = \emptyset$. Next, let us denote a subset of B by B_s, and let $G(B_s:\emptyset)$ be the graph obtained from G by contracting all the branches of B_s and let $G(\emptyset:B_s)$ be the graph obtained from G by deleting all the branches of B_s. Also, let $\Re(B_s)$ be the union of B_s and the set of self-loops of $G(B_s:\emptyset)$ and let $\eta(B_s)$ be the union of B_s and the set of bridges of $G(\emptyset:B_s)$. Moreover, if B is covered by successive application of operations $\Re$ and η to some subset $B_{(0)}$ of B, i.e., if

$$F_\ell(F_{\ell-1}(\ldots(F_1(B_{(0)}))\ldots)) \supseteq B$$

where $F_j (j = 1, \ldots, \ell)$ is either $\Re$ or η, then the ordered sequence of operations $\Re$ and η is called $\Re \cdot \eta$ sequence, and $B_{(0)}$ is called a generating branch set of B with respect to the $\Re \cdot \eta$ sequence.

With these definitions, we have the following theorem:

Theorem 1: The necessary and sufficient condition for H_{11} to be a triangular matrix is that $B(H_{11})$ is a generating branch set of B with respect to some $\Re\cdot\eta$ sequence.

Proof: Let $B_{(0)} = B(H_{21}) = B - B(H_{11})$ and let us assume that $B_{(0)}$ is a generating branch set of B with respect to the $\Re\cdot\eta$ sequence, i.e., let us assume that

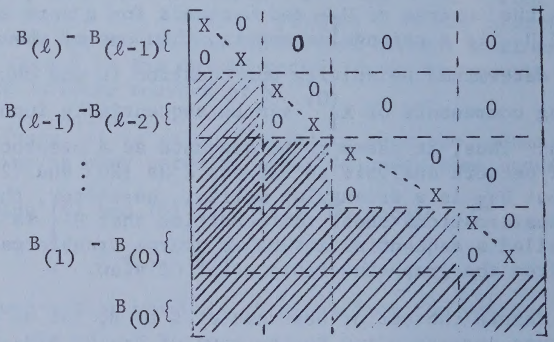
(a) $F_j (j = 1, \ldots, \ell)$ is either $\Re$ or η:

(b) $F_\ell(F_{\ell-1}(\cdot\cdot(F_1(B_{(0)}))\cdot\cdot)) \supseteq B$; and

(c) For each $j (j = 1, \ldots, \ell)$, $F_j(B_{(j-1)}) = B_{(j)}$, $B_{(j)} - B_{(j-1)} \neq \phi$, $B_{(\ell-1)} \not\supseteq B$, $B_{(\ell)} \supseteq B$.

Then, if F_j is η, each branch of $B_{(j)} - B_{(j-1)}$, together with some or all branches of $B_{(j-1)}$, defines an elementary cutset of G. If F_j is $\Re$, each branch of $B_{(j)} - B_{(j-1)}$, together with some or all branches of $B_{(j-1)}$, defines an elementary tieset of G. Hence, if we arrange the rows of $H_1 = \begin{bmatrix} H_{11} \\ \cdots \\ H_{21} \end{bmatrix}$ in the order of the branches of $B_{(\ell)} - B_{(\ell-1)}$, $B_{(\ell-1)} - B_{(\ell-2)}, \ldots, B_{(1)} - B_{(0)}$ and $B_{(0)}$, and further if we arrange the columns of H_1 in order of the cutsets or tiesets defined by branches of $B_{(\ell)} - B_{(\ell-1)}$, $B_{(\ell-1)} - B_{(\ell-2)}, \ldots$, and $B_{(1)} - B_{(0)}$, then H_1 has the form of

$$H_1 = \begin{bmatrix} H_{11} \\ \cdots \\ H_{21} \end{bmatrix} = \tag{22}$$



where the mark X denotes nonzero element and the hatched part means that zeros and nonzeros are distributed appropriately. Hence, if $B(H_{21}) = B - B(H_{11})$ is a generating branch set of B with respect to some $\Re\cdot\eta$ sequence, then H_{11} becomes a triangular matrix.

Conversely, it easily follows from the above statements that if H_{11} is a triantular matrix, then $B(H_{21}) = B - B(H_{11})$ is a generating branch set of B with respect to some $\Re\cdot\eta$ sequence.

Q. E. D.

Example 1: Consider a network N shown in (b) of Fig. 3,

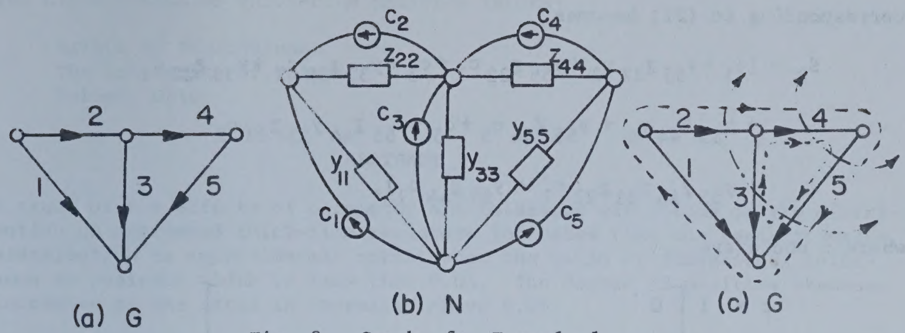


Fig. 3. Graphs for Example 1.

where $B = \{1,2,3,4,5\}$. Now, let $B_{(0)} = \{5\}$. Then, $B_{(1)} = \eta(B_{(0)}) = \{4,5\}$, $B_{(2)} = \Re(B_{(1)} = \{3,4,5\}$, $B_{(3)} = \eta(B_{(2)} = \{2,3,4,5\}$ and $B_{(4)} = \Re(B_{(3)}) = \{1,2,3,4,5\} = B$. Hence, $B_{(0)}$ becomes a generating branch set of B with respect to $\Re\cdot\eta$ sequence.

The elementary tieset defined by the branch of $B_{(4)} - B_{(3)} = \{1\}$ and some or all branches of $B_{(3)}$ becomes $\{1,2,4,5\}$, the elementary cutset defined by the branch of $B_{(3)} - B_{(2)} = \{2\}$ and some or all branches of $B_{(2)}$ becomes $\{2,3,5\}$, the elementary tieset defined by the branch of $B_{(2)} - B_{(1)} = \{3\}$ and some or all branches of $B_{(1)}$ becomes $\{3,4,5\}$ and the elementary cutset defined by the branch of $B_{(1)} - B_{(0)} = \{4\}$ and some or all branches of $B_{(0)}$ becomes $\{4,5\}$. These cutsets and tiesets are shown in (c) of Fig. 3.

Thus, H_1 becomes of the form:

$$H_1 = \begin{bmatrix} H_{11} \\ \hdashline H_{21} \end{bmatrix} = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ -\mathbf{z}_{22} & 1 & 0 & 0 \\ 0 & -y_{33} & 1 & 0 \\ -\mathbf{z}_{44} & 0 & -\mathbf{z}_{44} & 1 \\ \hdashline -1 & -y_{55} & -1 & -y_{55} \end{array} \right] \tag{23}$$

and the equation corresponding to (18) becomes

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -\mathbf{z}_{22} & 1 & 0 & 0 \\ 0 & -y_{33} & 1 & 0 \\ -\mathbf{z}_{44} & 0 & -\mathbf{z}_{44} & 1 \end{bmatrix} \begin{bmatrix} J_1 \\ U_1 \\ J_2 \\ U_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ {}_{22}c_2 \\ c_3 \\ {}_{44}c_4 \end{bmatrix} \tag{24}$$

Solving (24) sequentially only by substitution, we have $J_1 = c_1$, $U_1 = \mathbf{z}_{22}c_2 + \mathbf{z}_{22}c_1$, $J_2 = c_3 + y_{33}\mathbf{z}_{22}c_1 + y_{33}\mathbf{z}_{22}c_2$, and $U_2 = \mathbf{z}_{44}c_4 + \mathbf{z}_{44}c_3 + \mathbf{z}_{44}y_{33}\mathbf{z}_{22}c_1 + \mathbf{z}_{44}y_{33}\mathbf{z}_{22}c_2 + \mathbf{z}_{44}c_1$. Therefore, the equation corresponding to (21) becomes

$$\hat{S}_2 = [c_1 + y_{55} z_{22} c_2 + y_{55} z_{22} c_1 + c_3 + y_{33} z_{22} c_2 + y_{33} z_{22} c_1$$

$$+ y_{55} B_{44} c_4 + y_{55} z_{44} c_3 + c_5 + y_{55} z_{44} y_{33} z_{22} c_2$$

$$+ y_{55} z_{44} y_{33} z_{22} c_1 + y_{55} z_{44} c_1],$$

where $\hat{c}$ and $\hat{e}$ are

$$c = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \left[\begin{array}{l} 0 \\ c_2 + c_1 \\ -(y_{33} z_{22} c_2 + y_{33} z_{22} c_1) \\ c_4 + c_3 + c_1 + y_{33} z_{22} c_2 + y_{33} z_{22} c_1 \\ c_5 + c_3 + c_1 + y_{33} z_{22} c_2 + y_{33} z_{22} c_1 \end{array} \right]$$

and

$$e = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \left[\begin{array}{l} 0 \\ z_{22} c_2 + z_{22} c_1 \\ -(z_{22} c_2 + z_{22} c_1) \\ z_{44} c_4 + z_{44} c_3 + z_{44} y_{33} z_{22} c_2 + z_{44} y_{33} z_{22} c_1 + z_{44} c_1 \\ -(z_{22} c_2 + z_{22} c_1 + z_{44} c_3 + z_{44} y_{33} z_{22} c_2 + z_{44} y_{33} z_{22} c_1 + z_{44} c_1) \end{array} \right]$$

respectively.

As is easily deduced from the above example, a sequential method of source transformations can also be implemented by graphical manipulation.

REFERENCES

1. M. E. Van Valkenburg, "Network Analysis," 3rd edition, Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

2. M. F. Moad, "A Sequential Method of Network Analysis," IEEE Trans. on Circuit Theory, Vol. CT-17, No. 1, pp. 99-104, 1970.

THE DISTRIBUTION OF THICK-FILM RESISTOR VALUES*

Arthur R. Thorbjornsen
The University of Toledo
Toledo, Ohio

ABSTRACT

A study of the effects of parameter and tolerance variations on the distribution of untrimmed thick-film resistors indicates that the resistor value distribution is approximately normal when the ratio of dimensional tolerance to resistor width is less than 0.05. The degree of positive skewness increases as the ratio is increased above 0.05.

INTRODUCTION

Research is in progress on the computer-aided tolerance design of thick-film hybrid integrated circuits which has a goal of eliminating or reducing the amount of resistor trimming in the thick-film fabrication process [1]. Since Monte Carlo analysis is used in the above work, it is essential that a realistic distribution of untrimmed thick-film resistor values can be generated. Film-type resistors and some integrated devices are known to have a nonnormal, skewed distribution of values [2,3], but it is also claimed that a normal distribution can be assumed for such devices [4]. This paper presents the results of a theoretical and experimental study of how thick-film resistor value distributions are affected by variations in the tolerance and nominal value of width, length, and aspect ratio and shows that under certain conditions the distribution may be assumed to be Gaussian.

THE PROBABILITY DENSITY FUNCTION

The value of an ideal film-type resistor is given by

$$R = r_s L/W ,$$

which is a nonlinear function of the sheet resistivity r_s, the resistor length L, and the width W. These three parameters are assumed to be independent, continuous, normal, random variables. The length and width can safely be assumed to be independent because each of these dimensions is determined by a separate screen printing operation. Although the sheet resistivity (or film thickness) is actually dependent in a nonlinear way upon length and width [5,6,7], it is assumed, for simplicity, that there is no linear correlation between the sheet resistivity and the other parameters.

An expression for the probability density function for resistor values can be derived since the function and the parameter density functions are known. A nonlinear function of normal density functions will produce a nonnormal density function. Since the function has three parameters, the resistor probability density function involves a double integral [8], which is difficult to evaluate even using numerical techniques. Therefore, in this study a Monte Carlo approach was used instead. The method used was to generate

---

* This work was supported by NSF Grant GK-42076.

sets of independent, normally distributed values for L, W, and r_s and then to compute the resulting set of resistor values. A frequency distribution histogram of the set of resistor values approximates the probability density function.

THE COMPUTER EXPERIMENT

Numerous sets of ideal film-type resistor values were computer generated using various values of aspect ratio, resistor width, and dimensional tolerance ($\Delta W = \Delta L$). The generated resistor value distributions were then tested to see if they departed significantly from a normal or Gaussian distribution. The test statistics used were

$$\beta_1 = \frac{(\text{3rd moment about the mean})^2}{(\text{variance})^3} ,$$

which gives the degree of relative skewness, and

$$\beta_2 = \frac{(\text{4th moment about the mean})}{(\text{variance})^2} ,$$

which gives the degree of relative kurtosis [9]. A normal distribution has a $\beta_1 = 0.0$ and a $\beta_2 = 3.0$ [10].

For this experiment a sample size of 500 was chosen. The levels of significance for the statistical tests were selected to be 0.02 for β_1 (a one-tailed test) and 0.01 for β_2 (a two-tailed test). Therefore, if a computed β_1 is equal to, or greater than 0.065, we can conclude that the density function has a significant skewness, and if a computed β_2 is equal to or greater than 3.60 or equal to, or less than, 2.57 we can conclude that the kurtosis is significantly different from the normal $\beta_2 = 3.0$ [10]. The values for the experimental parameters were chosen to be ±10% for the sheet resistivity tolerance; 5/1 and 1/3 for the aspect ratio; 50.0, 20.0, and 10.0 mils for the resistor width; and ±5.0, ±3.0, ±2.0, ±1.0, ±0.5, and ±0.3 mils for the dimensional tolerance. For purposes of generating sets of parameter values, the percent tolerance, or percent variability, is assumed to be equal to 300 times the normalized parameter standard deviation. For each combination of parameter values, two sets of 500 resistor values were generated and the corresponding values of β_1 and β_2 generated. The results of this experiment are tabulated in Table 1. Figure 1 consists of a plot of values of β_1 versus the ratio $\Delta W/W$. From this figure we see that if the ratio of $\Delta W/W$ is less than about 0.05 all of the computed β_1 values are less than the critical value of 0.065. Thus, when $\Delta W/W > 0.05$ film-type resistors cannot be accurately simulated with a normal distribution. From Table 1 we see that the skewness is more pronounced at a high aspect ratio than at a low aspect ratio, since the critical value of β_1 is exceeded in 10 cases for L/W = 5/1 and only in 7 cases for L/W = 1/3.

A laboratory experiment was conducted to show that the skewed resistor value distributions do, in fact, exist. Forty-six thick-film resistor arrays were fabricated having the pattern shown in Figure 2. The array consists of 5 pairs of resistors all of which have an aspect ratio of 5/1. The resistor pairs have widths of 50.0, 30.0, 20.0, 10.0 and 6.67 mils. The resistors were printed on the one inch square substrates, the resistor values measured, and the values of β_1 and β_2 calculated. The calculated statistics are listed in Table 2.

For a sample size of 50 and a level of significance of 0.02 the critical value of β_1 is 0.619. Estimates of the critical values for β_2 at a level of significance of 0.01 are 2.0 and 5.0 [10]. The results in Table 2 show that the resistor distributions are significantly skewed when the resistor width is 20.0 or smaller. The resistor dimensions at which the resistor distribution becomes skewed will depend on the manufacturing equipment and processes used and may vary from one facility to the next. Figure 3 shows the frequency distribution histograms of two of the resistors fabricated in this experiment. The histogram in Figure 3(a) is for the 50 mil wide resistor, R_2, and does not appear to be skewed. The severely skewed histogram in Figure 3(b) is for the 10 mil wide resistor, R8.

CONCLUSIONS

Computer and laboratory experiments show that untrimmed thick-film resistors have skewed resistor value distributions. Under certain conditions, namely when the dimensional tolerance is insignificant with respect to the resistor width ($\Delta W/W < 0.05$), the resistor value distribution is approximately normal. In many cases, however, resistors will be constructed which do have skewed distributions. Therefore, when performing Monte Carlo analyses of integrated circuits, it is very likely that the simulation of integrated resistors using normal distributions will produce misleading results.

REFERENCES

[1] A. R. Thorbjornsen, "Computer-Aided Tolerance Assignment for Microelectronic Circuits," NSF Grant GK-42076, April 1, 1974.

[2] F. Jensen, "Component Data for Statistical Circuit Analysis," Microelectronics and Reliability, v. 11, pp. 79-85, February 1972.

[3] H. R. Camenzind, Circuit Design for Integrated Electronics. Reading, Mass.: Addison-Wesley, 1968, pp. 124-130.

[4] L. J. Ragonese, "A Circuit-Design Approach for Enhancing the Yield of Complex Iterative Integrated Electronic Subsystems," Proc. IEEE, pp. 1613-1616, December 1964.

[5] W. G. Dryden, R. J. Ost, and R. B. Wolf, "Characterization of Thick Film Resistives for Maximum Production Yields," Proc. 1970 Electronic Components Conf., pp. 412-418, May 1970.

[6] C. F. Jefferson, "Geometry Dependence of Thick Film Resistors," Proc. 1970 Electronic Components Conf., pp. 209-215, May 1970.

[7] D. E. Riemer, "The Effect of Geometry on the Characteristics of Thick Film Resistors," Proc. 1970 Electronic Components Conf., pp. 102-108, May 1970.

[8] G. J. Hahn and S. S. Shapiro, Statistical Models in Engineering. New York: Wiley, 1967, pp. 188-190.

		L/W = 5/1		L/W = 1/3	
W (mils)	ΔW (mils)	β_1	β_2	β_1	β_2
50.0	5.0	0.0330	2.882	0.0217	3.143
"	"	0.0279	2.716	0.1357*	2.922
"	3.0	0.0301	2.903	0.1326*	3.552
"	"	0.0001	2.616	0.0037	2.844
"	2.0	0.0020	3.033	0.0359	2.958
"	"	0.0172	2.652	0.0145	3.112
"	1.0	0.0136	2.762	0.0045	2.679
"	"	0.0151	3.265	0.0017	2.937
"	0.5	0.0190	2.758	0.0123	2.690
"	"	0.0005	2.938	0.0292	2.918
"	0.3	0.0025	2.749	0.0052	2.999
"	"	0.0026	3.374	0.0006	2.936
20.0	5.0	0.4750*	3.710*	0.0493	2.754
"	"	0.1840*	3.661*	0.0172	2.731
"	3.0	0.0952*	3.254	0.0076	2.826
"	"	0.1452*	3.455	0.0014	3.066
"	2.0	0.0153	3.039	0.0039	3.345
"	"	0.0265	3.064	0.0773*	2.938
"	1.0	0.0630	3.324	0.0005	2.767
"	"	0.0614	2.784	0.0015	2.758
"	0.5	0.0010	2.935	0.0241	2.794
"	"	0.0315	3.172	0.0032	2.858
"	0.3	0.0010	2.936	0.0023	2.978
"	"	0.0048	2.872	0.0076	2.614
10.0	5.0	1.6065*	5.666*	0.2600*	3.714*
"	"	0.7840*	3.945*	0.8388*	6.063*
"	3.0	0.1657*	3.093	0.0115	2.941
"	"	0.4323*	3.624*	0.0685*	3.080
"	2.0	0.1378*	3.300	0.0198	3.142
"	"	0.1469*	3.665*	0.1488*	3.815*
"	1.0	0.0322	2.889	0.0053	2.771
"	"	0.0200	2.706	0.0342	2.782
"	0.5	0.0222	3.289	0.0338	2.729
"	"	0.0001	2.990	0.0002	2.836
"	0.3	0.0021	3.352	0.0142	3.488
"	"	0.0134	2.837	0.0002	3.067

* Critical value exceeded.

Table 1. Results from computer experiment.

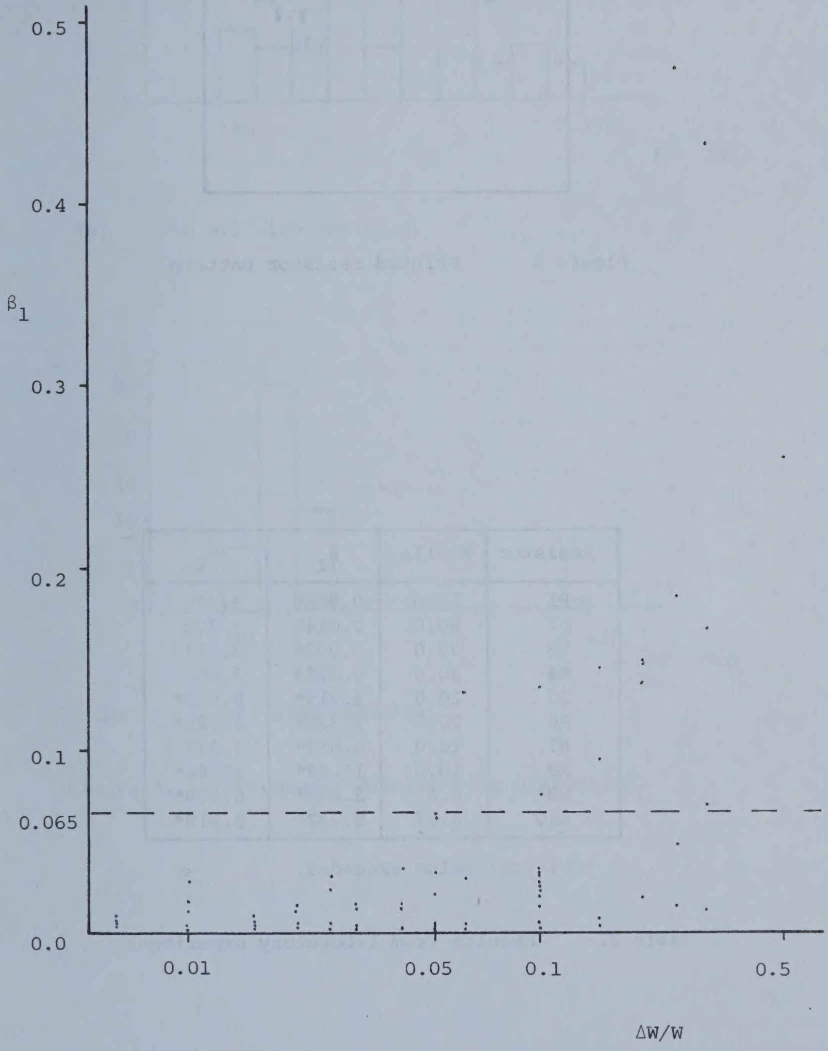


Figure 1. Results from computer experiment.

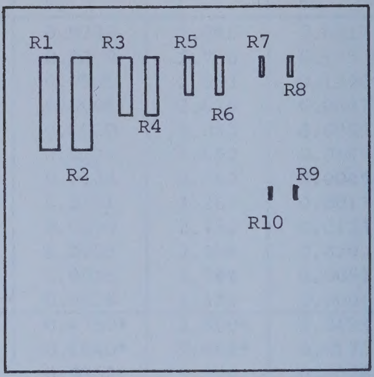
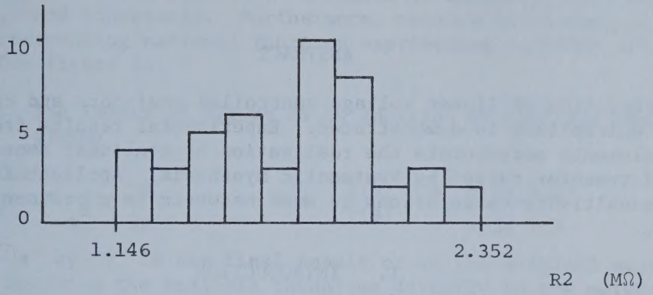


Figure 2. Printed resistor pattern.

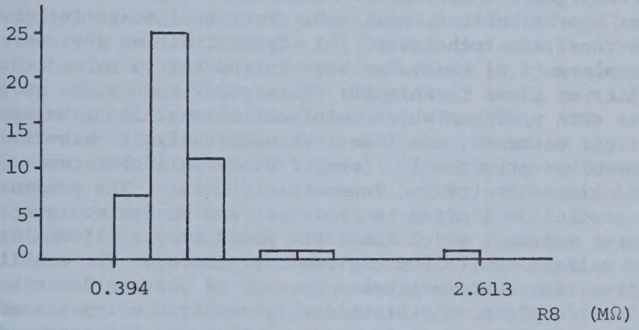
Resistor	W(mils)	β_1	β_2
R1	50.0	0.0886	3.294
R2	50.0	0.0148	2.322
R3	30.0	0.0006	2.517
R4	30.0	0.0184	3.068
R5	20.0	3.219*	8.503*
R6	20.0	1.331*	5.529*
R7	10.0	0.857*	3.217
R8	10.0	11.68*	17.44*
R9	6.67	2.675*	6.609*
R10	6.67	5.737*	8.918*

*Critical value exceeded.

Table 2. Results from laboratory experiment.



(a) 50 mil wide resistor.



(b) 10 mil wide resistor.

Figure 3. Frequency distribution histograms.

EXPERIMENTAL RESULTS WITH LINEAR CONTROLLED RESISTANCE AND CONDUCTANCE

N. R. MALIK and G. L. JACKSON
Division of Information Engineering, The University of Iowa,
Iowa City, Iowa 52242

ABSTRACT

Construction of linear voltage controlled resistors and conductors using IC multipliers is demonstrated. Experimental results from a network of such elements corroborate the realization of nonlinear amnesic driving point and transfer curves by systematic synthesis. Applicability of linear network sensitivity calculations to such networks is experimentally verified.

I. INTRODUCTION

In many applications it is necessary that an amnesic network be designed which has a specified nonlinear driving point (dp) or transfer curve (tc). Such applications include: compensation for temperature, pressure or other environmental factors, device modelling [1], harmonic and difference frequency generators, wave shaping, construction of nonlinear reactive elements [2,3], and analog models for studying complex nonlinear systems [4]. The only general approach to this kind of problem has involved piecewise linear approximations, nonlinear circuit elements (diodes), and piecewise linear analysis techniques. [4] Special active devices such as rotators and scalars [5,6] have also been introduced to extend the range of applicability of these techniques. This paper introduces an alternative approach to such problems which involves rational function approximations, linear circuit elements, and linear network analysis methods. The resulting networks contain only linear resistors, linear controlled resistors (LCR's) and linear controlled conductors (LCG's). The present experimentation is explicitly limited to resistors and conductors controlled by a network input voltage, which restricts the direct applicability of the results to voltage controlled dp and tc curves. The results demonstrate for the first time that nonlinear dp and tc curves described by certain rational functions can be systematically realized using present IC technology. The technique is analogous to well-known RLC synthesis. Sensitivities and component tolerances can be easily computed, and no graphical work is necessary. The method provides a novel alternative to use of nonlinear elements, and is complementary to this technique in many of its features.

II. MOTIVATION FOR THE EXPERIMENTS

The central idea of this paper is the analogy which exists between a class of linear, dynamic networks and a class of linear amnesic networks which possess nonlinear dp and tc curves. Figure 1a shows a Laplace-transformed R, $\pm L$, $\pm C$ network of arbitrary topology with all inductors and capacitors explicitly shown. Such networks are analyzed using linear algebraic equations involving the complex frequency variable s. Many properties of the network, including its terminal behavior, are described by rational functions in s. At least for positive components, systematic synthesis procedures are well-known. These start with a rational function approximation of the desired dynamic behavior and lead to a multitude of network structural options for realizing the given function. Figure 1b shows a network with topological structure presumed identical to that of

Figure 1a. The transformed inductors and capacitors, however, are replaced with voltage controlled linear resistors (VCR's) and voltage controlled conductors (VCG's) respectively. From Kirchhoff's laws and the device equations it follows that all analysis equations describing the network of Figure 1a also describe the network of Figure 1b except for the notational change $s \to v$, and conversely. Furthermore, network functions for Figure 1a induce corresponding rational function expressions in terms of the variable v for Figure 1b.

Example: The ladder of Figure 2a is analyzed by familiar techniques giving

$$\frac{V_o(s)}{V(s)} = \frac{-s}{s^2 - 3s + 3} \quad , \text{ and } \quad \frac{I(s)}{V(s)} = \frac{s^2 - 2s + 1}{s^2 - 3s + 3} \tag{1}$$

By replacing s by v in the final result or in the original equations, or by simply applying the analysis technique directly to the network of Figure 2b, one obtains

$$\frac{v_o}{v} = \frac{-v}{v^2 - 3v + 3} \quad , \text{ and } \quad \frac{i}{v} = \frac{v^2 - 2v + 1}{v^2 - 3v + 3} \tag{2}$$

Since numerators and denominators of rational functions such as (2) may be written in factored form, we see that R, VCR, VCG, dp, and tc functions may be represented by means of poles and zeros in a "complex v plane". Since the pole and zero values of (2) are the same as those of (1), which in turn are related to the structure and kinds of components of the network of Figure 2a, we see that we may effectively bring our insight and experience concerning RLC networks to bear on R, VCR, VCG networks. Partial and continued fraction expansions and other detailed expressions familiar in synthesis of networks of Figure 1a also apply to Figure 1b, except for notation. It follows that some R, VCR, VCG networks can be systematically synthesized by: 1) approximating the given dp or tc curve with a rational function in v, 2) testing the function against realizability conditions and repeating the approximation procedure until these conditions are satisfied, and by 3) synthesizing the network by familiar techniques. Necessary conditions for realizability of R, VCR, VCG networks have not as yet been determined, however some sufficient conditions have been previously described. [7]

The function

$$i(v) = \frac{v(v - 1)^2}{v^2 - 3v + 3} \tag{3}$$

was selected to demonstrate the synthesis of an R, VCR, VCG network. This particular function was chosen primarily because it produces the "interesting" dp curve of Figure 3a while not requiring ungrounded controlled devices. The pole-zero plot of (3) is shown in Figure 3b. This figure suggests that the dp curve of Figure 3a may be viewed as the intersection of the magnitude of the complex function (3) with a vertical plane through the real voltage axis. In fact, (3) was actually obtained by "thinking up" a pole-zero plot which would produce a tunnel diode-like curve for a dp plot!

III. EXPERIMENTAL RESULTS

To verify the theoretical predictions and test the practicality of the concepts a VCR, a VCG and an R, VCR, VCG network were constructed and tested. In this work only straightforward designs were used, and no efforts were made to optimize either the devices or the circuits.

Figure 4a shows how a grounded VCG can be constructed from a resistor and a four-quadrant multiplier. The multiplier is described by

$$v_o = K_1 v_i \bar{v} \quad . \tag{4}$$

Assuming the multiplier input impedance is high,

$$i_i = v_i \frac{(1 - K_1 \bar{v})}{R_i} \tag{5}$$

Figure 6a shows the experimental verification of Eq. (5) obtained from the circuits of Figure 5. The output of the circuit of Figure 5a was the input v_i of the circuit of Figure 5b. The input $\bar{v}$ of the circuit of Figure 5b was derived from a variable DC source. The two outputs of a Signetics 5595 four-quadrant multiplier are described approximately by

$$v_{out} = 20 \pm 0.05 \, \bar{v} \, v_i \tag{6}$$

The 2N2605 PNP transistor amplified and level-shifted one output of the 5595 so the circuit of Figure 5b would realize Eq. (4).

As expected, i_i varied linearly with v_i from $v_i = -10$ volts to $v_i = +10$ volts. The input voltage was swept at a 10 KHz rate. The curves showed no parasitic effects until the sweep rate reached 0.1 MHz, where phase distortion appeared as shown in Figure 6b.

The voltage offset in Eq. (5) is eliminated by providing additional circuitry which relates the multiplier input $\bar{v}$ to the control input v according to

$$\bar{v} = \frac{1 \pm Kv}{K_1} \quad . \tag{7}$$

Substituting (7) into (5) leads to the desired device characteristic

$$\frac{i_i}{v_i} = G_i(v) = \mp \frac{K}{R_i} v \quad . \tag{8}$$

Figure 7a shows the complete VCG circuit which realizes (8) and which was used in subsequent experiments. Observe that an amplifier is used to control the coefficient of v in (8), with both magnitude and sign easily adjustable by proper amplifier design. (The zener diodes and 10 K resistor at the v_y input were inserted for protection of the multiplier and do not affect circuit performance.) Figure 7b shows a VCR which was designed from the idealized circuit of Figure 4b. From Figure 4b,

$$v_i = R_i i_i + K_1 K_2 i_i \bar{v} \tag{9}$$

where K_2 is the conversion constant of a current to voltage transducer (difference amplifier and input resistor). An offset adjustment similar to (7) gives the result

$$\frac{v_i}{i_i} = R_i(v) = \pm KR_i v \tag{10}$$

Now a network containing VCR and VCG elements will be described. Equation (3) describes the network shown in Figure 2b. Since impedance scaling of RLC circuits induces impedance scaling of the corresponding R, VCR, VCG circuits, scaling all resistances both controlled and independent by 10 KΩ resulted in the experimental network shown in Figure 8. This network has the dp characteristic

$$i(v) = \frac{1}{10^4} \frac{v(v-1)^2}{v^2 - 3v + 3} \tag{11}$$

Figure 9a Curve A shows the experimental verification of (11). In spite of the fact that the two controlled resistors had negative values for positive values of v, no spurious oscillations were observed, and no trimming of the circuit of Figure 8 with reactive elements was required. The failure of the minimum to occur at precisely one volt is attributed to the use of 5% resistors in the adder circuit which senses v. This produced a scaling of the independent variable, an effect which is discussed later.

The voltage gain function

$$\frac{v_o}{v} = \frac{-v}{v^2 - 3v + 3} \tag{12}$$

was also computed for the circuit. Experimental verification is shown in Figure 9b.

Since the circuit operation was experimentally found to be rather sensitive to changes in R_2 (Figures 2b and 8), the sensitivity of the double zero in Eq. (11) to changes in R_2 was computed using the classical formula for multiple roots. [8] This calcualtion gives

$$\Delta v = \pm \sqrt{\frac{2\Delta R_2}{R_2}} \tag{13}$$

where Δv is the change in the zero position from its original value when R_2 is changed from its nominal value by an amount ΔR_2. Figure 3b and Eq. (12) suggest that a positive ΔR_2 will lead to a pair of simple zeros on the real axis, and that a negative ΔR_2 will produce complex zeros. Figure 9a Curves B and C show experimental verification of these results.

A practical design problem would involve assigning component tolerances so that the actual network dp function is sufficiently close to the prescribed one. For this purpose a sensitivity function of the form

$$S_{R_2}^{i(v)} = R_2 \frac{di}{dR_2} \tag{14}$$

would be useful. For the experimental network this function is

$$S_{R_2}^{i(v)} = \frac{1}{10000} \frac{-2v}{(v^2 - 3v + 3)^2} \quad . \tag{15}$$

Equation (15), plotted in Figure 10, is seen to be in general agreement with the results of Figure 9a Curves B and C.

A significant deterioration in circuit performance was noted at sweep frequencies exceeding 5KHz. Since the objectives of the experiments had been accomplished, no attempts were made to either improve the frequency response or to look for networks with better sensitivity functions than (15).

The observation that Eqs. (11) and (12) and all the other images of network functions have identical poles is readily predictable from RLC theory. The significance of this result is that none of the currents or voltages in the networks will increase without limit, as long as the poles (which can be determined in advance) are not real. The authors' initial fears that for some v a controlled resistor might assume the negative of the value of the "Thevenin resistance" it sees (e.g., in Figure 2b) and produce a short circuit across a nonzero voltage were thus proved groundless.

Another useful technique of linear analysis, frequency scaling, also has an analogous result in R, LCR, LCG networks. The analog operation is a scaling of the independent variable v by a factor r_o when all controlled elements are scaled by the impedance factor r_o and all independent resistors are left unscaled.

Equation (3), which describes the network of Figure 2b, is of the form

$$i(v) = vy(v) \qquad (16)$$

Multiplying every VCR(VCG) coefficient by r_o $(1/r_o)$ gives

$$y_{fs}(v) = \frac{i_{fs}(v)}{v} = y(r_o v) \qquad (17)$$

where subscript fs denotes the "frequency scaled" network. Equation (17) implies that any reference points such as peaks or valleys in $y(v)$ are shifted by the factor $1/r_o$. Equation (17) shows that the dependent variable is also scaled by this operation through the factor v. This amplitude scaling of the dependent variable is easily avoided in dp problems by also impedance scaling by the factor r_o. For impedance scaling,

$$y_{is}(v) = \frac{1}{r_o} y(v) \quad , \qquad (18)$$

where subscript "is" denotes impedance scaling. Figure 11 shows experimental verification of the scaling operation denoted by Eq. (17) for $r_o = 2$. This independent variable scaling was implemented by simply reducing the gain of the input amplifier (Figure 8) by a factor of 2. Notice that a change in the vertical and horizontal sensitivity of the scope was used to compensate for the r_o multiplicative factor in (17). It should be noted that in circuits constructed from nonlinear elements scaling of the independent variable is a difficult task requiring a special device called a scalar. [3]

The problem of devising ungrounded LCR and LCG devices does not appear to be exceedingly difficult. Availability of such devices would make it possible to utilize for dp and tc synthesis the multitude of network structures: Foster, Brune, Cauer, Darlington and all of the other well-known networks including mixed forms. Of course active RC techniques may be used to construct active R, VCR networks with very general properties.

IV. SUMMARY

Experimental verification of a new technique was described which provides an alternative method of realizing prescribed nonlinear dp and tc curves. This technique features analytic rather than graphical analysis and design techniques, simple sensitivity computations, a wide variety of possible network structures, and possibilities of decomposing complex problems into simple subproblems. Disadvantages of the new technique include more expensive components, applicability limited to functions, and these of a class not entirely described as yet, and a more difficult approximation technique. The advantages and disadvantages of the new technique are seen to be roughly complementary to those of the more conventional solution methods. The experimental results suggest that these techniques might lend themselves to practical implementation after some further development work.

REFERENCES

1. L. O. Chua, "Modeling of three terminal devices: A black box approach," IEEE Trans. Circuit Theory, Vol. Ct-19, pp. 555-562, November 1972.
2. R. W. Newcomb and O. A. Seriki, "Nonlinear capacitor simulation," Electron. Lett., Vol. 7, pp. 452-454, August 1971.
3. L. O. Chua, Introduction to Nonlinear Network Theory, McGraw-Hill, 1969.
4. L. O. Chua, "Synthesis of nonlinear systems with prescribed singularities," IEEE Trans. Circuit Theory, Vol. CT-18, pp. 375-382, May 1971.
5. L. O. Chua, "The rotator - a new network component," Proc. IEEE, Vol. 55, pp. 1566-1577, September 1967.
6. L. O. Chua, "Linear transformation convertor and its application to the synthesis of nonlinear networks," IEEE Trans. Circuit Theory, Vol. CT-17, pp. 584-594, November 1970.
7. Y. S. Kim, Synthesis of Nonlinear Driving-Point and Transfer Functions by Nonlinear Controlled Sources, Ph.D. Dissertation, The University of Iowa, December 1972.
8. L. P. Huelsman, Theory and Design of Active RC Circuits, McGraw-Hill, 1968, pp. 28-32.

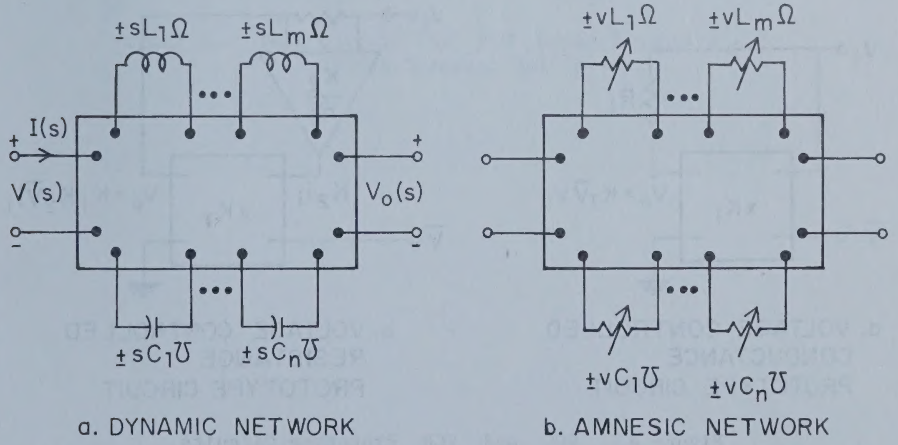
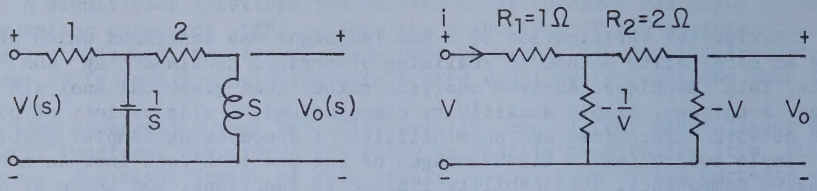


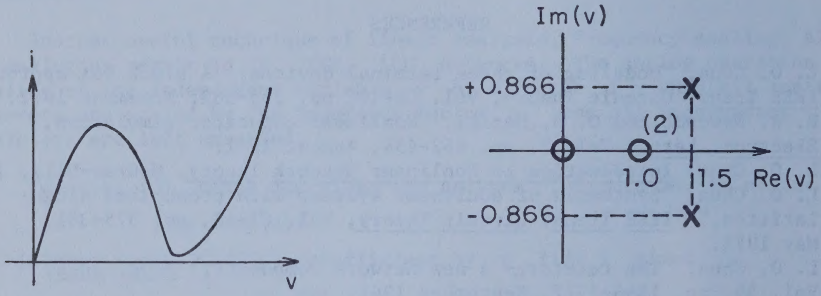
Figure 1. Dynamic R, ±L, ±C Network and its R, VCR, VCG Analog.


+
1
2
+
V(s)
$-\frac{1}{S}$
S
$V_o(s)$
-
-
i
$R_1=1\Omega$
$R_2=2\Omega$
+
+
V
$-\frac{1}{V}$
-V
V_o
-
-

a. RLC LADDER

b. R, VCR, VCG LADDER

Figure 2. Dynamic RLC Ladder and its R, VCR, VCG Analog.

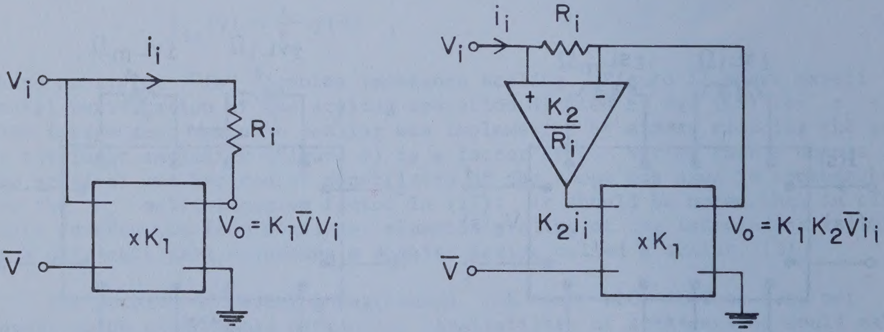


a. EXPECTED dp CURVE FOR Eq. (2)

b. POLE-ZERO PLOT OF Eq. (2)

Figure 3. Driving Point Curve and Pole-Zero Plot for Equation

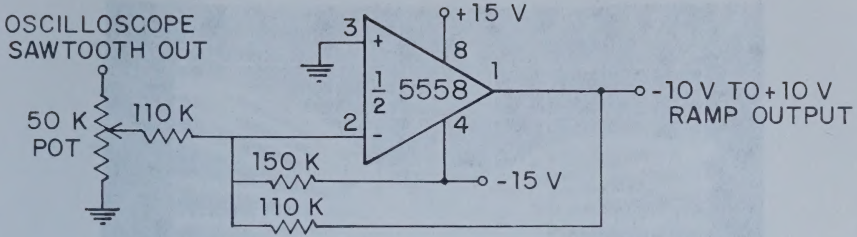
$$\frac{i}{v} = \frac{(v - 1)^2}{v^2 - 3v + 3} .$$


i_i
V_i
R_i
xK_1
$V_o = K_1\overline{V}V_i$
$\overline{V}$
i_i
R_i
V_i
+
-
$\frac{K_2}{R_i}$
$K_2 i_i$
xK_1
$V_o = K_1K_2\overline{V}i_i$
$\overline{V}$

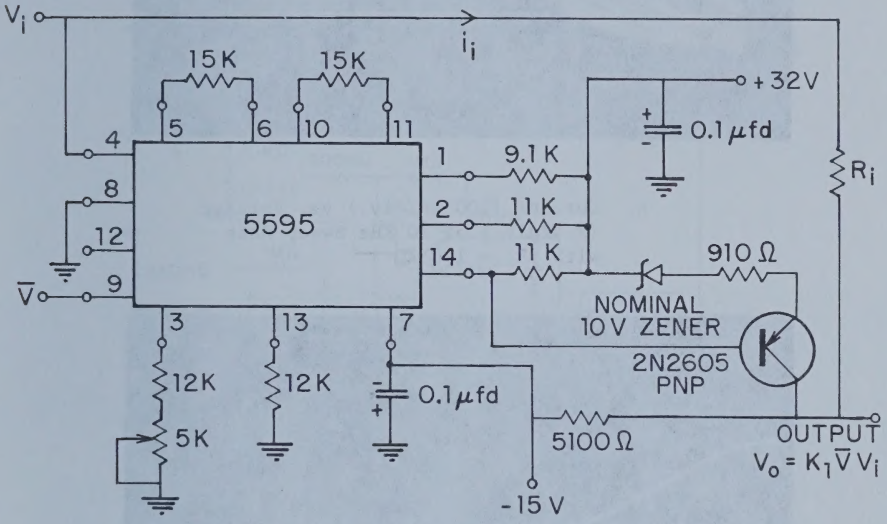
a. VOLTAGE CONTROLLED CONDUCTANCE PROTOTYPE CIRCUIT

b. VOLTAGE CONTROLLED RESISTANCE PROTOTYPE CIRCUIT

Figure 4. VCG and VCR Prototype Circuits.

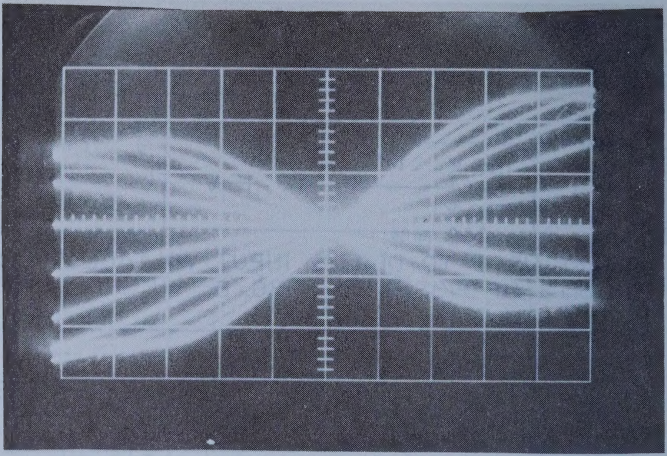

OSCILLOSCOPE
SAWTOOTH OUT
50 K
POT
110 K
+15 V
3
+
8
1
$\frac{1}{2}$ 5558
2
-
4
-10 V TO +10 V
RAMP OUTPUT
150 K
-15 V
110 K

a. CIRCUIT USED TO DERIVE INPUT VOLTAGES FOR CIRCUITS OF FIG. 5b AND FIG. 8.

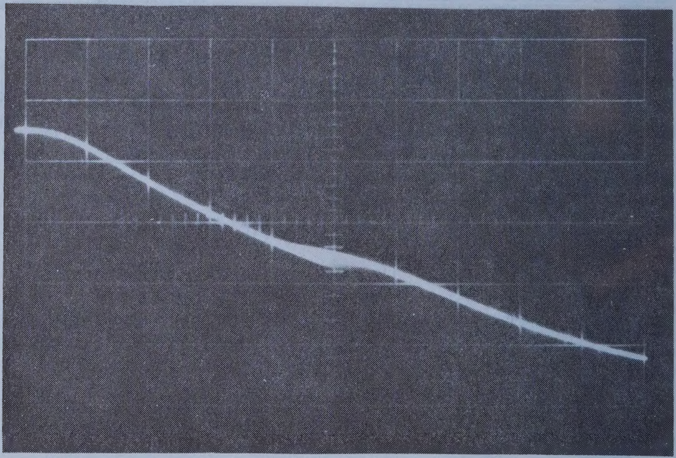

V_i
i_i
15K
15K
5
6
10
11
4
8
12
9
$\overline{V}$
5595
+32V
0.1 μfd
1
9.1 K
2
11 K
14
11 K
R_i
910 Ω
NOMINAL
10 V ZENER
2N2605
PNP
3
13
7
12K
12K
5K
0.1 μfd
5100 Ω
OUTPUT
$V_o = K_1 \overline{V} V_i$
-15 V

b. EXPERIMENTAL REALIZATION OF VCG PROTOTYPE CIRCUIT OF FIG. 4a.

Figure 5. Test Circuit for VCG Using Signetics 5595 4-Quadrant Multiplier.

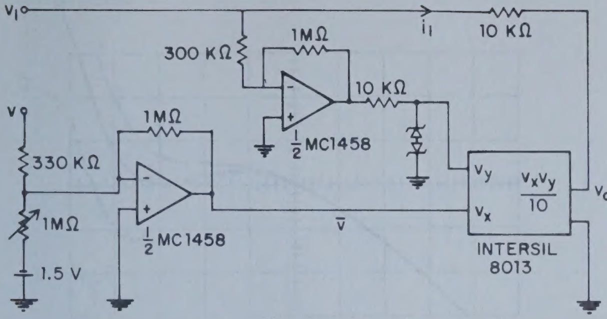


a. Current (100 μA/div.) vs. Voltage (2 V/div.) at 10 KHz Sweep Rate with R_i = 150 KΩ

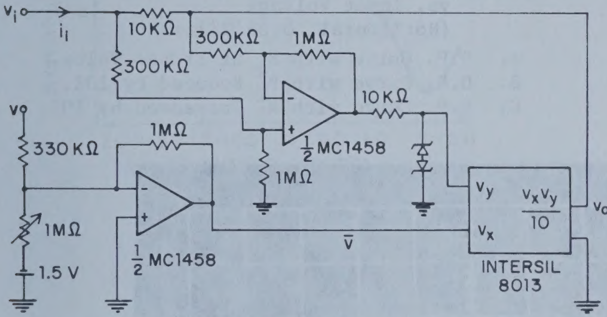


b. Current (100 μA/div.) vs. Voltage (2 V/div.) at 100 KHz Sweep Rate. $K_1\bar{v}$ = +3 volts.

Figure 6. Experimental VCG Driving Point Curves.


v₁
i₁
10 KΩ
1MΩ
300 KΩ
10 KΩ
½MC1458
v
1MΩ
330 KΩ
1MΩ
1.5 V
½MC1458
v̄
Vy
Vx
VxVy/10
Vo
INTERSIL 8013

a. VOLTAGE CONTROLLED CONDUCTANCE CIRCUIT


v₁
i₁
10KΩ
300KΩ
1MΩ
300KΩ
10KΩ
½MC1458
1MΩ
v
1MΩ
330KΩ
1MΩ
1.5 V
½ MC1458
v̄
Vy
Vx
VxVy/10
Vo
INTERSIL 8013

b. VOLTAGE CONTROLLED RESISTANCE CIRCUIT

Figure 7. Experimental VCG and VCR Circuits.

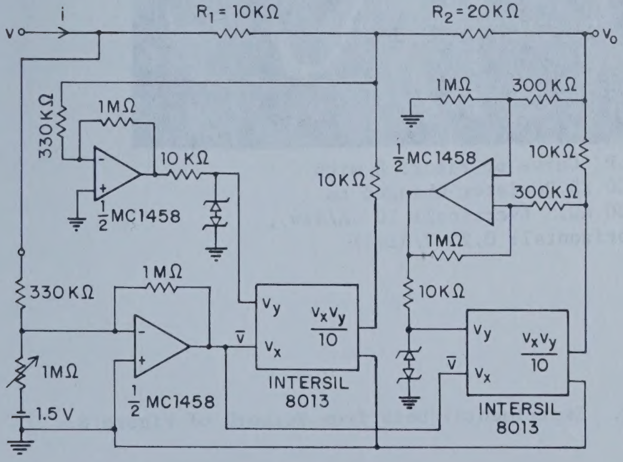
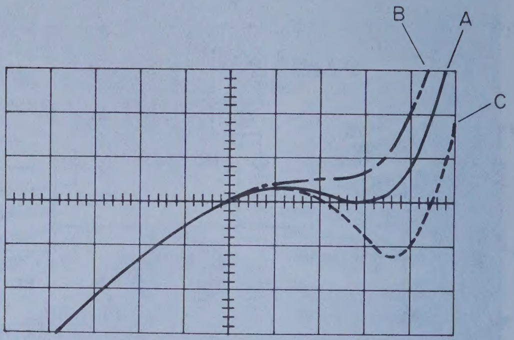
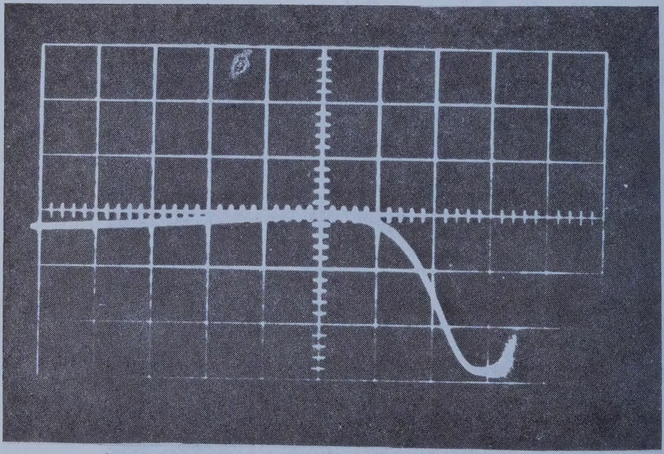

i
v
R₁ = 10KΩ
R₂ = 20KΩ
Vo
330KΩ
1MΩ
10 KΩ
½MC1458
1MΩ
300 KΩ
10KΩ
½MC1458
10KΩ
300KΩ
1MΩ
10KΩ
1MΩ
330KΩ
1MΩ
1.5 V
½ MC1458
v̄
Vy
Vx
VxVy/10
INTERSIL 8013
v̄
Vy
Vx
VxVy/10
INTERSIL 8013

Figure 8. Experimental Realization of Equation 3.



a. Driving Point Current
(Vertical, 20 μA/div.)
vs. Input Voltage
(Horizontal, 0.25 V/div.)

A. D.P. Curve with R_2 at Proper Value.
B. D.P. Curve with R_2 Reduced by 10%.
C. D.P. Curve with R_2 Increased by 10%.



b. D.P. Curve of Figure 8 with
330 KΩ Resistor Changed to
220 KΩ. (Vertical: 20 μA/div.,
Horizontal: 0.25 V/div.)

Figure 9. Experimental Data from Network of Figure 8.

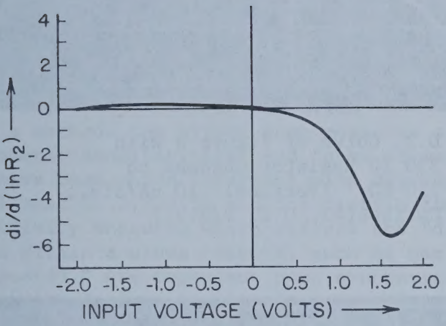
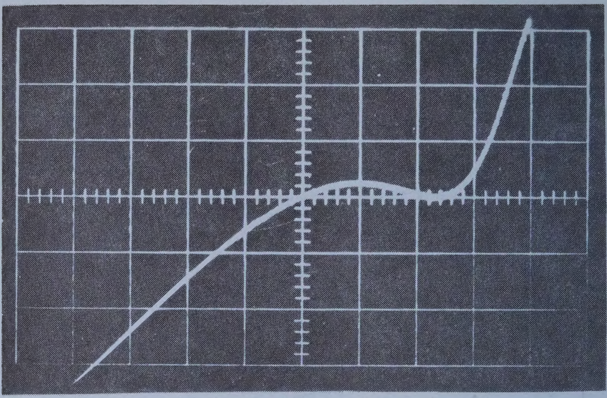
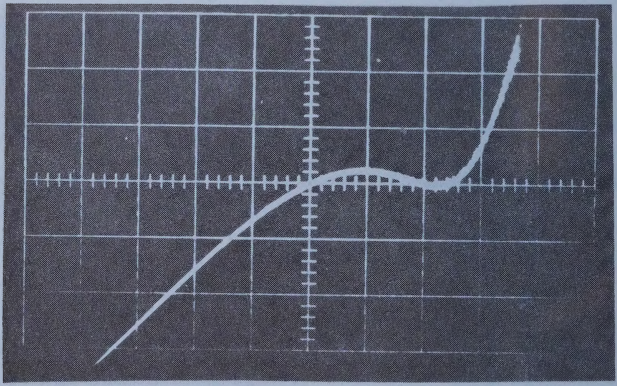


Figure 10. Sensitivity of Input Current to Value of R_2.



a. D.P. Curve of Figure 8 with 330 KΩ Resistor Changed to 110 KΩ. (Vertical: 10 μA/div., Horizontal: 0.25 V/div.)



b. Output Voltage V_o (Vertical, 1V/div.) vs. Input Voltage (Horizontal, 0.25 V/div.)

Figure 11. Scaling of Dependent Variable.

DISTRIBUTED PARAMETER SENSITIVITY: O'SHEA, WYNDRUM AND A NEW DP SYNTHESIS METHOD

JAMES K. LINN
Sandia Laboratories, Albuquerque, New Mexico 87115

SHLOMO KARNI
The University of New Mexico, Albuquerque, New Mexico 87131

ABSTRACT

Synthesis forms affect some sensitivity measures in Uniform RC Distributed Networks. Wyndrum and O'Shea synthesis methods for URC distributed networks are compared from a sensitivity standpoint. A new decomposition form for synthesis is shown to result in a network with reduced parameter sensitivity.

INTRODUCTION

Synthesis methods can affect some measures of parameter sensitivity. "Sums of parameter sensitivities" and "sums of magnitudes of parameter sensitivities" have been shown invariant with driving-point (DP) synthesis methods for lumped [1,2,3,4,5] and for distributed parameter [1,6,7] networks. Sensitivity measures which reflect the "distribution" of parameter sensitivities within a given network, such as the "sum of the sensitivity magnitudes squared," are dependent upon synthesis method. Techniques for optimizing network designs to minimize sums of sensitivity magnitudes squared have been developed also [8,9]. The fundamental question remains: "Which basic synthesis methods yield "less sensitive" networks, where the measure of sensitivity is dependent on synthesis method?"

Here the O'Shea and Wyndrum synthesis methods for URC distributed networks are compared with a new decomposition method, based on minimizing the sum of the sensitivity magnitudes squared.

The sum of parameter sensitivity magnitudes squared is minimized in a given network when all similar parameters have equal sensitivity. This leads to the concept of a network "sensitivity distribution." In many cases, an equal or balanced sensitivity distribution is optimal.

Synthesis techniques under investigation are analyzed from a sensitivity distribution standpoint by considering: (1) the bounds on overall distribution imposed by the synthesis method topology, and (2), the distribution of parameter sensitivity within the basic subsection one-ports, each one-port being formed by two or four URC distributed sections (for Wyndrum and O'Shea synthesis methods).

The concept of sensitivity redistribution or "sensitivity migration" with frequency and transform plane pole and zero locations is also used to analyze sensitivity distributions in URC distributed networks.

I. Sensitivity Distribution as Dictated by General Synthesis Methods

Historically, several general methods have been developed to calculate parameter sensitivities for lumped-parameter networks. These include adjoint networks [10], Thevenin equivalents [11], auxiliary networks [12], and others [13,14]; they obtain the sensitivity of a function (driving-point or transfer) with respect to a one-port lumped-parameter impedance or admittance embedded within the network.

We can easily extend these results to distributed networks. From the classical definition of parameter sensitivity we have

$$S^{F}_{e_i} = S^{F}_{Z_i} \; S^{Z_i}_{e_i} \tag{1}$$

where F is a network function, Z_i a one-port impedance, and e_i is a parameter of Z_i. General techniques are used to calculate $S^{F}_{Z_i}$ and we can use the classical definition of sensitivity to obtain $S^{Z_i}_{e_i}$, i.e.,

$$S^{Z_i}_{e_i} = \frac{e_i}{Z_i} \; \frac{\partial Z_i}{\partial e_i} \tag{2}$$

where Z_i is an impedance realized by distributed sections.

Consider the general synthesis forms which result in series decompositions of one-port networks (these may contain multiple distributed URC sections). Synthesis forms of this type include the Foster I and II forms of Wyndrum T-plane ($\tanh\sqrt{RCs}$ plane) DP synthesis methods, and the O'Shea P-plane ($\cosh\sqrt{RCs}$) forms for DP immittances [15].

For example, a series synthesis for a URC driving-point impedance $Z_i(s)$ can be expressed as

$$Z_i(s) = Z_1(s) + \ldots + Z_k(s) + \ldots + Z_n(s) \tag{3}$$

where the $Z_k(s)$, k=1,2,...,n are one-port DP impedances composed of either one, two or four URC distributed networks. This places basic constraints on the network sensitivity of the one-ports. From Eq. (1) we have

$$S^{Z_i(s)}_{e_k} = S^{Z_i(s)}_{Z_k(s)} \; S^{Z_k(s)}_{e_k} \tag{4}$$

Using Eq. (3) we obtain

$$S^{Z_i(s)}_{e_k} = \frac{Z_k(s)}{Z_i(s)} \; S^{Z_k(s)}_{e_k} \tag{5}$$

Each $Z_k(s)$ contains up to four URC distributed sections; hence the number of parameters in each one-port impedance, $Z_k(s)$, is of the same order (within a factor of four).

The sum of the sensitivity magnitudes in the k^{th} one-port impedance can be expressed from Eq. (5) as

$$\sum_{j=1}^{m} \left| S^{Z_i(s)}_{e_j} \right| = \left| \frac{Z_k(s)}{Z_i(s)} \right| \sum_{j=1}^{m} \left| S^{Z_k(s)}_{e_j} \right| \tag{6}$$

where m is the number of parameters in $Z_k(s)$.

We know that the sum of the sensitivity magnitudes of $Z_k(s)$ is independent of the synthesis form [5], i.e.,

$$\sum_{j=1}^{m} \left| S^{Z_i(s)}_{e_j} \right| = \left| \frac{1}{Z_i(s)} \right| \Big(\mathrm{Re}\,(Z_k(s)) - \mathrm{Im}\,(Z_k(s)) \Big) \tag{7}$$

It has been shown [7] that the resistive and capacitive sensitivities, individually, must be equal for the various $Z_k(s)$; thus $\mathrm{Re}(Z_k(s))$ and $\mathrm{Im}(Z_k(s))$

must be constant for all k.

One obvious conclusion can be drawn from this result: Canonic series synthesis forms cannot have a totally balanced sensitivity distribution (all parameter sensitivies equal for all frequencies).

Although various synthesis forms may not provide a totally balanced sensitivity distribution, some forms are much closer to a balanced sensitivity distribution than others. These forms should be identified:

Let us consider the form of Wyndrum's Foster I impedance decomposition in the T-plane:

$$Z(T) = K_o T + \frac{K_\infty}{T} + \sum_i \frac{A_i T}{(T^2 + p_i^2)}, \quad T = \tanh \sqrt{R_{ij} C_{ij} s} \tag{8}$$

where K_∞, K_o, A_i, and p_i are nonnegative constants. The resulting section impedances in the T-plane are therefore

$$K_o T, \frac{K_\infty}{T}, \frac{A_i T}{(T^2 + p_i^2)} \quad \text{for } i = 1,2,\ldots$$

Conditions on T, and p_i which will obviously result in a very unbalanced sensitivity distribution are:

1. If $K_\infty \neq 0$ i.e., Z(T) has a pole at T=0, then for $|T| << 1$ and $|T| << p_i$, the impedance $|K_\infty/T|$ becomes large, and the sensitivities of its parameters unbalance the sensitivity distribution, i.e., become large relative to the other network parameters.

2. For $p_i >> |T|$ and $p_i >> p_j$ for any i,j, the resulting parallel section impedances will not be of the same order of magnitude; hence a very unbalanced sensitivity distribution will result.

This analysis applies equally to Foster II series admittance forms, i.e., $Y(T) = \Sigma Y_k(T)$; here the poles of Y(T), i.e., zeros of Z(T), are considered instead of the poles of Z(T) in conditions (1) and (2).

Cauer Synthesis Forms of one-ports can be similarly analyzed; a summary of results is noted here. As known, Cauer I synthesis forms of Z(T) and Cauer II forms of Y(T) result in short-circuit URC distributed sections for the parallel elements, i.e., $Z_k = (K_i/T)$, K_i constant. For $|T| << 1$, this results in large impedance magnitudes in the parallel elements and small impedance magnitudes in the series elements. The sensitivity distribution will tend to be more equal under these conditions.

For $|T|$ large (i.e., $|T| \simeq 1$) Cauer synthesis forms result in impedances of URC distributed sections of approximately equal magnitudes. The sensitivity of the overall impedance or admittance to parameters embedded further into a network is considerably less than for parameters near the ports. This necessarily results in an unbalanced sensitivity distribution.

II. Sensitivity Migration

In Section I, the overall sensitivity distribution was related to the basic synthesis form. The next step is an investigation into the sensitivity distribution within the one-ports.

The basic equivalence between impedances synthesized in the tanh-plane and in the cosh-plane is known; specifically, the cosh-plane synthesis requires twice the number of sections.

Consider the parallel network given by

$$Z_k(s) = \frac{A_k \tanh \sqrt{RCs}}{\sqrt{s}(\tanh^2 \sqrt{RCs} + b_k^2)} \quad , \quad A_k \geq 0 \tag{9}$$

O'Shea synthesizes this as

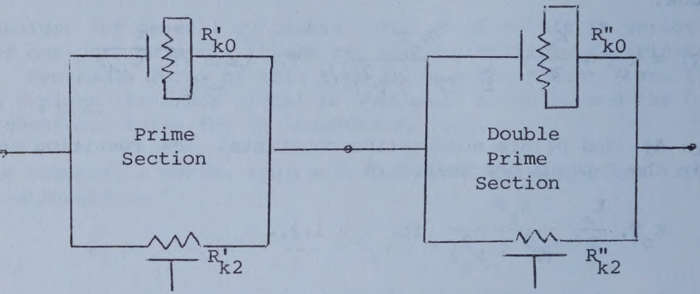


Figure 1. O'Shea Realization of Eq. (9).

for the k^{th} parallel section and using the designation of "0" and "2" type elements. The prime and double prime superscripts are to distinguish between the two parallel sections.

The value of the resistive parameters are [15]:

$$R'_{k0} = \frac{4d}{1 - x} \qquad\qquad R''_{k0} = \frac{4d}{1 + x}$$

$$R'_{k2} = \frac{d}{1 + x} \qquad\qquad R''_{k2} = \frac{d}{1 - x}$$

where

$$d = \frac{A_k}{2} \; \frac{RC}{(1 + b_k^2)} \qquad\qquad x = \frac{1}{(1 + b_k^2)^{1/2}}$$

The new variable "x" is the magnitude of the pole in the P-plane. (O'Shea's cosh $\sqrt{RCs}$ plane). Note that $0 \leq x \leq 1$.

The resistive sensitivity ratio of the two sections in the "prime" network can be obtained by differentiation. This result is given by

$$\frac{S_{R'_{k2}}^{Z_k(s)}}{S_{R'_{k0}}^{Z_k(s)}} = \frac{R'_{k2}(R'_{k0}(P + 1))^2 (P - 1)(1 + F/Q)}{R'_{k0}(R'_{k2}(P - 1))^2 (P + 1)(1 - F/Q)} \tag{10}$$

where

$$F = \sqrt{RCs} \; , \quad P = \cosh(F) \; , \text{ and } Q = \sinh(F)$$

Simplifying and considering magnitudes only, we obtain

$$\left| \frac{S^{Z_k(s)}_{R'_{k2}}}{S^{Z_k(s)}_{R'_{k0}}} \right| = \left| \left(\frac{(1+x)}{(1-x)}\right) \left(\frac{(P+1)}{(P-1)}\ \frac{(F+Q)}{(F-Q)}\right) \right| \tag{11}$$

The first term (1+x)/(1-x), is dependent only upon the pole position; hence it is designated as a "pole-position" factor. The second term is dependent only on frequency, (the RC product is constant); hence it is termed a "frequency" factor.

For small values of b_k, the pole position factor is given by

$$\lim_{b_k \to 0} \left| \frac{1+x}{1-x} \right| = \lim_{x \to 1} \left| \frac{1+x}{1-x} \right| = \infty \tag{12}$$

and, for large values of b_k

$$\lim_{b_k \to \infty} \left| \frac{1+x}{1-x} \right| = \lim_{x \to 0} \left| \frac{1+x}{1-x} \right| = 1 \tag{13}$$

Similarly, the limits of the frequency factor are

$$\lim_{|s| \to 0} \left| \frac{(P+1)}{(P-1)}\ \frac{(F+Q)}{(F-Q)} \right| = \infty \tag{14}$$

and

$$\lim_{|s| \to \infty} \left| \frac{(P+1)}{(P-1)}\ \frac{(F+Q)}{(F-Q)} \right| = 1 \tag{15}$$

It is clear that for the "prime" section, the magnitude of the ratio of resistive sensitivities becomes large at low frequencies or for small values of pole magnitudes in the T-plane.

Other resistive sensitivity magnitude ratios can be obtained. Figure 2 shows the overall sensitive distribution normalized to $S^{Z_k(s)}_{R'_{k2}}$.

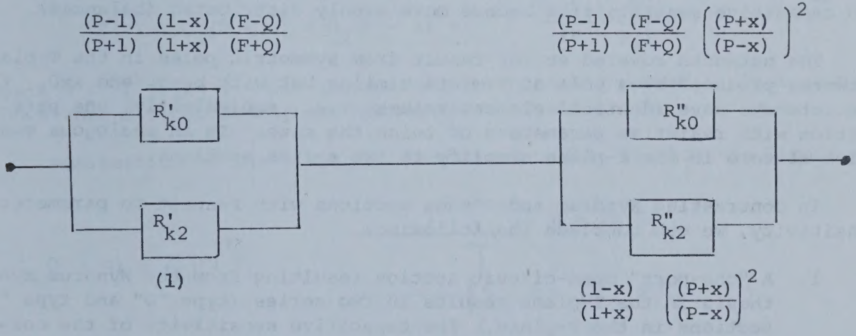
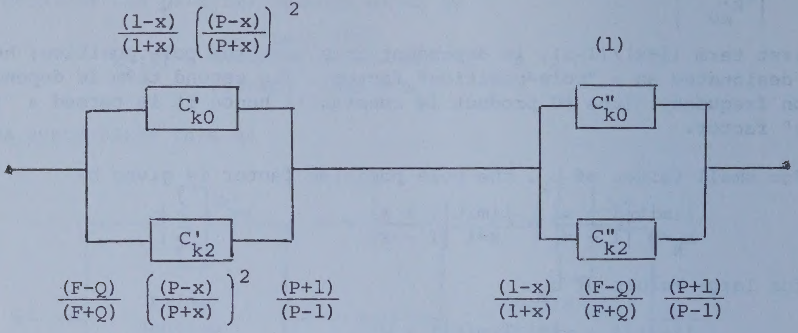

$\frac{(P-1)}{(P+1)}\ \frac{(1-x)}{(1+x)}\ \frac{(F-Q)}{(F+Q)}$
$\frac{(P-1)}{(P+1)}\ \frac{(F-Q)}{(F+Q)} \left(\frac{(P+x)}{(P-x)}\right)^2$
R'_{k0}
R'_{k2}
R''_{k0}
R''_{k2}
(1)
$\frac{(1-x)}{(1+x)} \left(\frac{(P+x)}{(P-x)}\right)^2$

Figure 2. Relative Resistive Sensitivity Magnitude Distribution.

Obviously, for sections synthesizing small pole magnitudes in the T-plane, i.e., when $x \simeq 1$, the resistive sensitivity will migrate to R'_{k2} and R''_{k0}. (Note, P is complex and, for $s=j\omega$, the term $|(P-x)|$ cannot go to zero except when P and x both go to zero.)

For low frequencies, the magnitude of the term (F-Q)/(F+Q) approaches zero, causing a migration of resistive sensitivity from the "0" type sections to the "2" type sections.

A similar capacitive sensitivity distribution, normalized by the sensitivity with respect to C''_{k0}, is shown in Figure 3.



$\frac{(1-x)}{(1+x)}\left(\frac{(P-x)}{(P+x)}\right)^2$
(1)
C'_{k0}
C''_{k0}
C'_{k2}
C''_{k2}
$\frac{(F-Q)}{(F+Q)}\left(\frac{(P-x)}{(P+x)}\right)^2\frac{(P+1)}{(P-1)}$
$\frac{(1-x)}{(1+x)}\frac{(F-Q)}{(F+Q)}\frac{(P+1)}{(P-1)}$

Figure 3. Relative Capacitive Sensitivity Magnitude Distribution

Again, a clearly defined sensitivity migration occurs for pole positions near zero in the T-plane, i.e., x near unity. In this case, the sensitivity migrates to C''_{k0} and C'_{k2}. Sensitivity migration with fequency is not as pronounced since

$$\lim_{|RCs|\to 0}\left|\frac{(F-Q)}{F+Q}\,\frac{(P+1)}{(P-1)}\right| \approx \frac{1}{3} \tag{16}$$

The net result for low frequencies is that the capacitive sensitivity migrates to C'_{k0} and C''_{k0}, although, in the limit, the ratio to C'_{k2} and C''_{k2}, respectively, is only about 3:1.

In summary, we have shown how the resistive sensitivity migrates for low frequencies and small pole values to R'_{k2}, and the capacitive sensitivity to C''_{k0}. For higher frequencies and larger pole magnitudes, the resistive and capacitive sensitivities become more evenly distributed (balanced).

The networks covered so far result from symmetric poles in the T-plane. Networks produced by a pole at T=∞ are similar but with $b_k=\infty$, and x=0. The two networks have identical element values, i.e., equivalently, one parallel section with resistive parameters of twice the size. In an analogous manner, poles at zero in the T-plane simplify to two series sections.

In contrasting Wyndrum and O'Shea sections with respect to parameter sensitivity, we can conclude the following:

1. A "one-port" open-circuit section resulting from the Wyndrum synthesis in the T-plane results in two series (type "0" and type "2") sections in the P-plane. The capacitive sensitivity of the one-port migrates to the "0" type element and the resistive sensitivity to the "2" type element for small pole values and lower frequencies. This, in essence, decouples the sensitivies of the open-circuit Wyndrum synthesis section.
2. A "one-port" short-circuit Wyndrum section is realized by parallel type "0" and type "2" sections in the P-plane. Again, the capacitive sensitivity of the one port migrates to the "0" type element and the resistive sensitivity to the "2" type element for small

pole values and lower frequencies, again, decoupling the short-circuit section parameter sections.

3. Parallel open-circuit and short-circuit sections resulting from Wyndrum synthesis techniques result in low series "parallel" sections in the P-plane. The resistive and capacitive sensitivities of the overall combination are concentrated in R'_{k2} and C''_{k0} respectively for small values of b_k and frequency.

4. O'Shea synthesis of DP impedances also realizeable by Wyndrum's synthesis methods, in essence, simply decouple into different sections the capacitive and resistive sensitivities. The sum of the sensitivity magnitudes squared for network realized by the two synthesis methods is approximately equal although one synthesis form employs twice the number of sections. The same conclusion applies to admittances since $S^{Z_i}_{e_k} = -S^{1/Z_i}_{e_k}$.

III. Alternate Decomposition of O'Shea Driving-Point Synthesis

In this section, a new decomposition of O'Shea's DP synthesis form is shown to simultaneously reduce parameter sensitivity at the cost of increased number of elements.

Any realizeable (by URC sections) DP impedance is written as [15]

$$Z(P) = K \frac{\prod_{i=1}^{n+1} (P - z_i)}{\prod_{i=1}^{n} (P - p_i)} , \quad P = \cosh \sqrt{RCs} \tag{17}$$

$K > 0$, p_i and z_i interlace on the real axis and

$$|z_i| \le 1, \quad |p_i| \le 1$$

Consider the decomposition of Z(P) given by

$$Z(P) = C_1(P + \alpha) + C_2(P - \alpha) + \prod_{i=1}^{n} \frac{A_i(P^2 - \alpha^2)}{(P - p_i)} \tag{18}$$

where $0 \le \max_i |z_i| \le \alpha \le 1.0$ (Note: $\alpha=1$ for O'Shea's method.) It is easily shown that C_1, C_2, and A_i are nonnegative for all i.

Realization of the first two terms in Eq. (18) is shown in Figure 4.

$C_1(P + \alpha) = Z_1(P) =$

$C_2(P - \alpha) = Z_2(P) =$

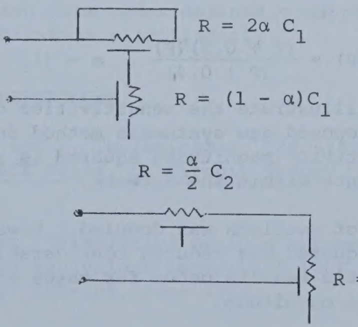


Figure 4. Realization of Terms of Eq. (18).

Other terms are parallel combinations of the one-ports.

Previously, it was shown that at low frequencies, both the O'Shea and Wyndrum DP synthesis methods result in poor overall impedance distributions and hence poor sensitivity distributions, i.e., nearly all capacitive and resistive parameter sensitivities in certain elements. The new decomposition results in a redistribution of the overall impedances and hence sensitivities. This redistribution (for $\alpha<1$) necessarily results in a reduction of the more sensitive parameters, yielding a network with a more balanced distribution.

Consider how the overall impedance is redistributed. The values of A_i are given by

$$A_i = \left. \frac{Z(P)\ (P - p_i)}{(p_i^2 - \alpha^2)} \right|_{P=p_i} \tag{19}$$

Since $\alpha > p_i$ for all i, then A_i increases with a decrease in α. Also, C_1 and C_2 can be expressed as

$$C_1 = \frac{1}{2\alpha} \left\{ \frac{(\alpha - z_1) \ \ldots. \ (\alpha - z_{n+1})}{(\alpha - p_1) \ \ldots. \ (\alpha - p_n)} \right\} \tag{20}$$

$$C_2 = -\frac{1}{2\alpha} \left\{ \frac{(-\alpha - z_1) \ \ldots. \ (-\alpha - z_{n+1})}{(-\alpha - p_1) \ \ldots. \ (-\alpha - p_n)} \right\} \tag{21}$$

where z_1 and z_{n+1} are the most negative and most positive zeros in the P-plane, respectively. As the value of α approaches z_{n+1}, C_1 will tend to zero, and simultaneously, α will approach $-z_{n+1}$. (Note: $(|z_{n+1}| = |z_1|)$. Thus, C_2 is driven to zero.) The overall relative impedance distribution has been shifted from the series sections (C_1 and C_2) to the parallel sections necessarily.

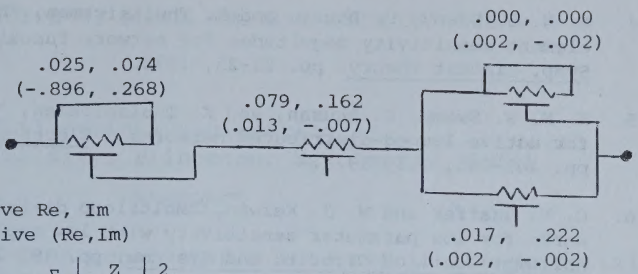
The net effect on sensitivity is a reduction of the sensitivity of the series sections and an increase in sensitivity of the parallel sections. Additionally, within the pair of sections synthesizing each of the first two terms, the sensitivity of the O'Shea sections is reduced to the factor α, the open-circuit section associated now with each O'Shea section assuming the sensitivity reduction of the O'Shea section.

To illustrate this point and the more equal sensitivity distribution potentially obtainable by this method, consider the sum of the sensitivity magnitudes squared as a measure of the sensitivity distribution. Let Z(P) be as follows:

$$Z(P) = \frac{(P + 0.5)(P)}{(P + 0.4)}, \quad s = j1, \quad RC = 1.0 \tag{22}$$

Figures 5 and 6 illustrate the sensitivities of the O'Shea synthesis method ($\alpha=1$) and the proposed new synthesis method for $\alpha=0.55$, respectively. The sum of the sensitivity magnitudes squared is also given as a measure of the sensitivity balance within the network.

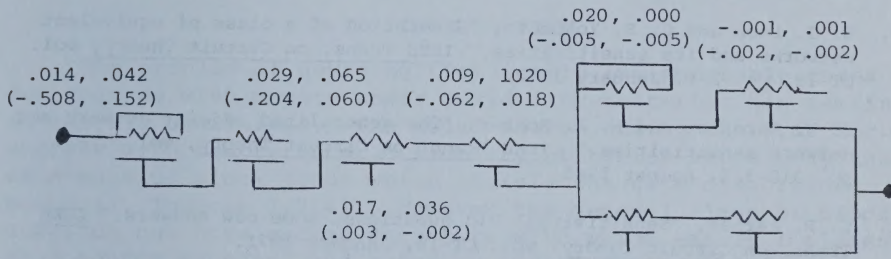
The number of sections was doubled. However, the sum of the sensitivity magnitudes squared was reduced considerably, by more than a factor of two. More dramatic results occur for cases where the largest and smallest zero are equal in magnitude.


.000, .000
(.002, -.002)
.025, .074
(-.896, .268)
.079, .162
(.013, -.007)
.017, .222
(.002, -.002)

Sensitivities: Resistive Re, Im
Capacitive (Re,Im)

$$\sum_i \left| S^Z_{e_i} \right|^2 = .9137$$

Figure 5. O'Shea Realization of Eq. (22).


.020, .000
(-.005, -.005)
-.001, .001
(-.002, -.002)
.014, .042
(-.508, .152)
.029, .065
(-.204, .060)
.009, 1020
(-.062, .018)
.017, .036
(.003, -.002)
.035, .060
(.005, -.003)
.018, .033
(-.106, .039)

Sensitivities: Resistive Re, Im
Capacitive (Re, Im)

$$\sum_i \left| S^Z_{e_i} \right|^2 = .3589$$

Figure 6. Proposed Alternate Realization of Eq. (22).

SUMMARY

The synthesis form for URC distributed networks affects the sensitivity distribution and, thus, some measures of sensitivity. In a comparison, the O'Shea synthesis method employed twice the number of URC sections of Wyndrum forms but under shown conditions, resulted in an equally poor sensitivity distribution. A new decomposition form, under defined conditions, was shown to improve the sensitivity distribution significantly.

REFERENCES

1. M. N. S. Swamy, C. Bhusan, and K. Thulasiraman, "Sensitivity invariants for nonlinear networks," IEEE Trans. on Circuit Theory, vol. CT-19, pp. 599-606, Nov. 1972.

2. J. V. Leeds and G. I. Ugron, "Simplified mutiple parameter sensitivity calculation and continuously equivalent networks," IEEE Trans. on Circuit Theory, vol. CT-14, pp. 188-191, June 1967.

3. A. G. J. Holt and J. K. Fidler, "Summed sensitivity of network functions," Electronic Letters, no. 4, pp. 85-87, 1968.

4. M. N. S. Swamy, C. Bhusan and K. Thulasiraman, "Bounds on the sum of element sensitivity magnitudes for network functions," Proc. IEEE Int. Symp. Circuit Theory, pp. 21-25, 1972.

5. M. N. S. Swamy, C. Bhusan, and K. Thulasiraman, "Sensitivity invariants for active lumped-distributed networks," Electronics Letters, vol. 8, pp. 441-445, July 1973.

6. C. V. Shaffer and W. J. Kerwin, "Multiloop distributed RC active networks for low parameter sensitivity with low amplitude gain," 2nd Asilomar Conf. on Circuits and Systems, pp. 197-201, 1968.

7. James K. Linn, "Parameter Sensitivity and Synthesis of Passive Uniform Distributed Networks," Ph.D. Dissertation, The University of New Mexico, Albuquerque, New Mexico, July 1974.

8. J. D. Schoeffler, "The synthesis of minimum sensitivity networks," IEEE Trans. on Circuit Theory, vol. CT-11, pp. 271-276, June 1964.

9. B. J. Leon and C. E. Yokomoto, "Generation of a class of equivalent networks and its sensitivities," IEEE Trans. on Circuit Theory, vol. CT-19, pp. 2-8, January 1972.

10. S. W. Director and R. A. Rohrer, "The generalized adjoint network and network sensitivities," IEEE Trans. on Circuit Theory, vol. CT-16, pp. 318-323, August 1965.

11. S. R. Parker, "Sensitivity: Old questions, some new answers," IEEE Trans. on Circuit Theory, vol. CT-18, January 1971.

12. J. V. Leeds, Jr., "Transient and steady-state sensitivity analysis," IEEE Trans. on Circuit Theory, vol. CT-13, pp. 288-289, 1966.

13. H. Bode, Network Analysis and Feedback Amplifier Design, Van Nostrand, New Jersey, 1945.

14. S. R. Parker, E. Peskin and P. M. Chirlian, "Application of a bilinear theorem to network sensitivity," IEEE Trans. on Circuit Theory, vol. CT-12, pp. 448-450, August 1965.

15. M. S. Ghausi and J. J. Kelley, Introduction to Distributed-Parameter Networks, Holt, Rinehart and Winston, New York, 1968.

TREE CODING OF STATIONARY BLOCK ERGODIC SOURCES WITH A FIDELITY CRITERION*

HARRY H. TAN
Department of Electrical Engineering
Princeton University, Princeton, New Jersey 08540

ABSTRACT

Recently there has been considerable interest in the potential of tree coding as a solution to the source coding implementation problem. In this paper we prove a tree coding theorem for abstract-alphabet stationary block ergodic sources under a single-letter fidelity criterion. The proof of this theorem uses recent results from the theory of branching processes with random environments.

I. INTRODUCTION

The problem of devising practically implementable codes for sources with memory under a fidelity criterion has remained largely unexamined. The usual source coding theorems for sources with memory under a fidelity criterion shows only the existence of block codes which satisfy the rate distortion bound [1, Theorem 7.2.4]. However the general class of block codes do not have enough structure with which easily implementable source encoding and decoding algorithms may be devised. Goblick [2] was the first to suggest that tree codes may be desirable for source coding under a fidelity criterion. Subsequently, the existence of tree codes which satisfy the rate distortion bound for discrete-time memoryless sources under a bounded single-letter fidelity criterion was proved by Jelinek [3]. More recently implementable tree encoding algorithms have been devised by Anderson and Jelinek [4] and by Gallager [5] applicable to the class of symmetric source-user pairs. Viterbi and Omura [6] have also demonstrated the applicability of the Viterbi algorithm for trellis encoding of memoryless discrete-time sources. These results, although valid only for memoryless sources, indicate the potential of tree coding schemes as a solution to the source coding instrumentation problem.

In light of the above results, a tree coding theorem valid for sources with memory under a fidelity criterion would be of considerable interest. The first result in this direction is Berger's tree coding theorem for autoregressive Gaussian sources under a square-error fidelity criterion valid in a region of low distortion values [1, Theorem 6.3.4]. Recently Gray [12] has proved a trellis coding theorem valid for time-discrete stationary ergodic sources under a fidelity criterion

---

* This research was supported by NSF Grant No. GK-42080.

generated by a metric distortion measure.

In this paper we prove a tree coding theorem for abstract-alphabet time-discrete stationary block ergodic sources under a per-letter fidelity criterion. A very recent paper by Davis and Hellman [7] has demonstrated that the application of extinction criteria for Galton-Watson branching processes [8, Chapter I] in Jelinek's proof [3] of the tree coding theorem is valid only for symmetric source-user pairs. Davis and Hellman [7] then reestablish Jelinek's tree coding theorem for memoryless sources by using the extinction criteria for branching processes with random environments due to Smith and Wilkinson [9]. The proof presented in this paper uses a more general extinction criteria for branching processes with random environments established by Athreya and Karlin [10].

II. THE TREE CODING THEOREM

We will use the standard notation in [1] and denote a discrete-time source with abstract alphabet A_o by $(A,\mathcal{A},\mu)$. Here A_o is assumed to be an abstract measurable space with σ-algebra $\mathcal{A}_o$, $A = \mathop{\times}_{t=-\infty}^{\infty} A_t$ where $A_t = A_o$ for each t, $\mathcal{A}$ is the σ-algebra of subsets of A generated by the cylinder sets and μ is a probability measure on $(A,\mathcal{A})$. We will assume that $(A,\mathcal{A},\mu)$ is strictly stationary and block-ergodic, that is, the shift transformation T on $(A,\mathcal{A},\mu)$ is measure preserving and that T^n is ergodic for each $n \geq 1$. Berger [14] has shown that the condition of block ergodicity is intermediate in restrictiveness between ergodicity and weak mixing [1, p.284]. Although block ergodicity is appreciably more restrictive than ergodicity, there are many sources with memory of interest which are block ergodic. For example, an irreducible aperiodic Markov chain is block ergodic. Let $X_t : A \to A_0$ denote the t-th coordinate function on A, that is, for $\omega \epsilon A$, $X_t(\omega) = \omega_t$. We note that X_t is the random variable which is the source output at time t, and we will regard the output of the source $(A,\mathcal{A},\mu)$ as the discrete-time stochastic process $\{X_t\}_{t=-\infty}^{\infty}$.

Let $A^n = \mathop{\times}_{t=o}^{n-1} A_t$, $\mathcal{A}^n = \mathop{\times}_{t=o}^{n-1} \mathcal{A}_t$ and μ_n the measure on $(A^n,\mathcal{A}^n)$ induced by μ. We will assume that the reproduction alphabet $\hat{A}_o$ is also an abstract measurable space with σ-algebra $\hat{\mathcal{A}}_o$. Define $(\hat{A},\hat{\mathcal{A}})$ and $(\hat{A}^n,\hat{\mathcal{A}}^n)$ analogous to $(A,\mathcal{A})$ and $(A^n,\mathcal{A}^n)$ respectively. In this paper we will consider a fidelity criterion $F_\rho = \{\rho_n : n \geq 1\}$ generated by a measurable distortion measure $\rho : A_o \times \hat{A}_o \to [o,\infty)$ according to

$$\rho_n(\underline{x},\underline{y}) = \frac{1}{n}\sum_{k=o}^{n-1} \rho(x_k,y_k) ,$$

for $\underline{x} = (x_o,\ldots,x_{n-1}) \in A^n$ and $\underline{y} \in \hat{A}^n$ similarly.

We refer to the source-user pair $(A,\mathcal{A},\mu)$ and F_ρ as a discrete-time abstract alphabet stationary block ergodic source under a single-letter fidelity criterion. In order to define the rate distortion function R(D) for this source-user pair, we need the following notation. For each $n \geq 1$, let $q_n : A^n \times \hat{\mathcal{A}}^n \to [0,1]$ be a conditional probability measure which induces a joint probability measure p_n on $\mathcal{A}^n \times \hat{\mathcal{A}}^n$ by

$$p_n(G) = \int q_n(x,G_x)\,d\mu_n(x) \quad , \; G \in \mathcal{A}^n \times \hat{\mathcal{A}}^n. \tag{1}$$

Also denote the marginal probability measure that p_n induces on $\hat{\mathcal{A}}^n$ by

$$\nu_n(F) = p_n(A^n \times F) \quad , \; F \in \hat{\mathcal{A}}^n . \tag{2}$$

Define the average mutual information $I_n(\mu_n,q_n)$ to be ∞ if $p_n \not\ll \mu_n \times \nu_n$ and to be $\int \log f_n \, dp_n$ if $p_n \ll \mu_n \times \nu_n$ where f_n is the R-N derivative. For each $D \geq 0$, let $Q_n(D)$ be the class of conditional probability measures q_n for which $\int \rho_n(x,y)\,dp_n(x,y) \leq D$. Define

$$R_n(D) = \inf_{q_n \in Q_n(D)} n^{-1} I(\mu_n,q_n) \quad , \tag{3}$$

and the rate distortion function R(D) by

$$R(D) = \lim_{n\to\infty} R_n(D) \quad . \tag{4}$$

The rate distortion function has the following operational significance in source coding applications. A block code of size K and blocklength n is a set $B = \{\underline{y}^{(1)},\ldots\ldots,\underline{y}^{(K)}\} \subset \hat{A}^n$. Such a code is said to have rate $R = \frac{\log K}{n}$. The code B is used to represent or encode the output of the source $(A,\mathcal{A},\mu)$ in the following manner. The source output $\{X_t\}_{t=-\infty}^{\infty}$ is segmented into a doubly-infinite sequence of n-blocks $\{\ldots\ldots,\underline{X}'_{-1},\underline{X}'_0,\underline{X}'_1,\ldots\ldots\}$ where $\underline{X}'_i = (X_{in},\ldots\ldots,X_{in+n-1})$ and each random n-block $\underline{X}'_i$ is mapped or encoded into the codeword $\underline{y} \in B$ which minimizes $\rho_n(\underline{X}'_i,\underline{y})$. For each $\underline{x} \in A_0^n$ and such code B, define $\rho_n(\underline{x}|B) = \min_{\underline{y} \in B} \rho_n(\underline{x},\underline{y})$. When the code B is used to encode the source $(A,\mathcal{A},\mu)$ an average distortion of $\rho_n(B) = \int \rho_n(\underline{x}|B)d\mu_n(\underline{x})$ is said to incur. The source coding problem is to determine the smallest rate R for which there exists a block code B with $\rho_n(B) \le D$. For the sources and fidelity criteria considered in this paper, the smallest rate is R(D) [11].

However coding performance closer to the rate distortion curve requires increasing code blocklength. Hence general block codes which require codebook encoding are impractical to implement. It is also known that tree codes have a structural dependence between codewords which can be used to facilitate the task of encoding the source. A tree code is a block code in which the codewords form a branching tree structure. A tree code B of depth ℓ with α branches per node and β letters per branch contains $K = \alpha^{\ell}$ codewords, each a sequence of $\ell\beta$ reproduction letters obtained by concatenating the letters on the ℓ branches of each unique path through the code tree. Such a code is said to have rate $R = \frac{\log \alpha}{\beta}$ and blocklength $\ell\beta$. In this paper we establish the following tree coding theorem.

Theorem: Consider a discrete-time stationary block ergodic source $(A,\mathcal{A},\mu)$ and a single-letter fidelity criterion F_{ρ} with associated rate distortion function R(D). Assume that there exists a $y \in \hat{A}_0$ such that

$$\int \rho(x,y)d\mu_1(x) < \infty . \tag{5}$$

Then for any $\epsilon > 0$ and $D \ge 0$ such that $R(D) < \infty$, there exists a tree code of sufficiently large blocklength and rate less than $R(D) + \epsilon$ which incurs an average distortion less than $D + \epsilon$ when used to encode the output of the source.

Proof: We will only provide a sketch of the proof here. All omitted details may be found in [13]. The proof is based, as usual, on a random coding argument. The general outline of the proof follows Jelinek's approach [3]. Fix an arbitrary $\epsilon > 0$ and a $D \ge 0$ such that $R(D) < \infty$. We want to show the existence of a tree code of rate less than $R(D) + \epsilon$ and average distortion less than $D + \epsilon$. From the definition of R(D) in (4), there exist integers α and β such that $R_{\beta}(D) + \frac{\epsilon}{4} \le \log \frac{\alpha}{\beta} < R(D) + \epsilon$. Under the assumption (5) on the distortion measure ρ, it was shown in [14] that the infinium in (3) can be restricted to those $q_n \in Q_n(D)$ which induce a

ν_n concentrated on a finite subset of A^n. Thus there exists a finite subset $\hat{Y} \subset \hat{A}^\beta$ such that if $R^*_\beta(D) = \inf_{q_\beta \in Q'_\beta(D)} \beta^{-1} I(\mu_\beta, q_\beta)$, then $R^*_\beta(D) < R_\beta(D) + \frac{\epsilon}{4}$, where $Q'_\beta(D)$ is the set of all $q_\beta \in Q_\beta(D)$ concentrated on $A^\beta \times \hat{Y}$. Moreover, there is a $q^*_\beta \in Q'_\beta(D)$ achieving the above infinium [14]. Let ν^*_β be the probability measure on $(\hat{A}^\beta, \hat{\mathcal{A}}^\beta)$ induced by q^*_β. Now for each integer $\ell \geq 1$ and $n = \ell\beta$, let $\mathcal{B}^n$ denote the class of all tree codes of depth ℓ with α branches per node and β reproduction letters per branch. We obtain a random ensemble of tree codes in $\mathcal{B}^n$ for each n in the following manner. A tree code is randomly chosen from $\mathcal{B}^n$ by independently selecting the block of β reproduction letters on each branch of the code tree according to the probability measure ν^*_β. Let λ_n be the probability measure on $\mathcal{B}^n$ which governs the above random code selection. The above steps leading to the use of ν^*_β to define the tree code ensemble is crucial since ν^*_β is concentrated on a finite set, thus in effect reducing ρ to a bounded distortion measure. Define $D^* = \inf_{\ell \geq 1} \bar{\rho}_{\ell\beta}$ where $\bar{\rho}_{\ell\beta}$ is the code ensemble average distortion. The main part of the of the proof is to show that $D^* < D + \frac{\epsilon}{2}$. Then we can pick an integer $\ell \geq 1$ sufficiently large enough so that $\bar{\rho}_{\ell\beta} \leq D + \epsilon$. Since the code ensemble average distortion is less than $D + \epsilon$, it follows that there must exist at least one tree code $B \in \mathcal{B}^{\ell\beta}$ for which $\rho_{\ell\beta}(B) \leq D + \epsilon$. Since all tree codes in $\mathcal{B}^{\ell\beta}$ have rate $R = \frac{\log \alpha}{\beta} < R(D) + \epsilon$, we would have established the theorem. The following lemma plays a similar role as in Jelinek's proof.

Lemma 1: For each $\eta > 0$, there exists an integer $\ell^* \geq 1$ such that for every $\ell \geq \ell^*$, $\lim_{k \to \infty} (\mu_{km} \times \lambda_{km})(\{(\underline{x}, B) : |\rho_{km}(\underline{x}|B) - D^*| > \eta\}) = 0$, where $m = \ell\beta$.

Let $\ell \geq 1$ be an arbitrary positive integer and let $\delta > 0$ be arbitrary. Segment the output $\{X_t\}_{t=-\infty}^{\infty}$ of the source into a doubly infinite sequence of $\ell\beta$ - blocks $\{\ldots\ldots, \underline{X}'_{-1}, \underline{X}'_0, \underline{X}'_1, \ldots\ldots\}$ where $\underline{X}'_i = (X_{i\ell\beta}, \ldots, X_{i\ell\beta + \ell\beta - 1})$. Define $D_{\ell\beta} : A^{\ell\beta} \to [o, \infty)$ by $D_{\ell\beta}(\underline{x}) = \frac{1}{\ell} \sum_{i=o}^{\ell-1} D_\beta(\underline{x}'_i)$ for $\underline{x} = (\underline{x}'_o, \ldots, \underline{x}'_{\ell-1})$, where $D_\beta(\underline{x}) = \int \rho_\beta(\underline{x}, \underline{y}) dq^*_\beta(\underline{y}|\underline{x})$. We can now define an integer-valued stochastic process $\{Z_m\}_{m=o}^{\infty}$ as follows. Let $Z_o = 1$ and for each $m \geq 1$, let Z_m = number of codewords $\underline{Y} = (\underline{Y}'_o, \ldots, \underline{Y}'_{m-1})$ in a randomly selected tree code $B^{m\ell\beta}$ such that $\rho_{\ell\beta}(\underline{X}'_i, \underline{Y}'_i) \leq D_{\ell\beta}(\underline{X}'_i) + \delta$ for all $i = 0, .., m-1$. It can be shown that the process $\{Z_m\}$ is a branching process with a stationary ergodic environmental sequence [10]. Using the extinction criteria developed in [10] the following lemma can be proved.

Lemma 2: For each integer $\ell \geq 1$ and each $\delta > 0$, let $q = \lim_{m\to\infty} (\mu_{m\ell\beta} \times \lambda_{m\ell\beta})(\{Z_m = 0\})$. Then there exists a $\delta \epsilon (0, \frac{\varepsilon}{2})$ and an integer $\ell_1 \geq 1$ such that $q < 1$ for every integer $\ell \geq \ell_1$.

Using lemmas 1 and 2 and the individual ergodic theorem, it can be shown that $D^* < D + \frac{\varepsilon}{2}$, which completes the proof of the theorem.

III. SUMMARY

In this paper we have presented a tree coding theorem for abstract-alphabet time-discrete stationary block ergodic sources under a single-letter fidelity criterion. Our theorem is weaker than Gray's result [12] in that block ergodicity is more restrictive than ergodicity and that trellis codes are a subclass of tree codes. However we are able to treat a large class of distortion measures and slightly more general source alphabets than Gray [12]. We note that Anderson and Jelinek [4] and Gallager [5] have devised and analyzed implementable tree encoding algorithms for symmetric sources. The analysis in [4] and [5] use the theory of ordinary branching processes. We feel that using the more powerful tools of branching processes with random environments as in this paper, the results in [4] and [5] may be extended to sources with memory.

REFERENCES

[1] T. Berger, Rate Distortion Theory - A Mathematical Basis for Data Compression, Englewood Cliffs, N.J., Prentice Hall 1971.

[2] T.J. Goblick, Jr., "Coding for a discrete information source with a distortion measure", Ph.D. dissertation, Dept. of Elec. Engrg., M.I.T., Cambridge, Mass., October 1962.

[3] F. Jelinek, "Tree encoding of memoryless time-discrete sources with a fidelity criterion", IEEE Trans. Inform. Theory, Vol. IT-15, pp. 584-590, September 1969.

[4] J.B. Anderson and F. Jelinek, "A 2-cycle algorithm for source coding with a fidelity criterion", IEEE Trans. Inform. Theory, Vol. IT-19, pp. 77-91, January 1973.

[5] R.G. Gallager, "Tree encoding for symmetric sources with a distortion measure", IEEE Trans. Inform. Theory, Vol. IT-20, pp. 65-76, January 1974.

[6] A.J. Viterbi and J. Omura, "Trellis encoding of memoryless discrete-time sources with a fidelity criterion", IEEE Trans. Inform. Theory, Vol. IT-20, pp. 325-332, May 1974.

[7] C.R. Davis and M.E. Hellman, "On tree coding with a fidelity criterion", submitted for publication.

[8] T.E. Harris, The Theory of Branching Processes, New York, Springer-Verlag, 1963.

[9] W.L. Smith and W.E. Wilkinson, "On branching processes in random environments", Ann. Math. Stat., Vol. 40, pp. 814-827, June 1969.

[10] K.B. Athreya and S. Karlin, "On branching processes with random environments: I. Extinction probabilities", Ann. Math. Stat., Vol. 42, pp. 1499-1520, October 1971.

[11] T. Berger, "Rate distortion theory for sources with abstract alphabets and memory", Information and Control, Vol. 13, pp. 254-273, 1968.

[12] R.M. Gray, "Time-invariant trellis encoding of ergodic discrete-time sources with a fidelity criterion", submitted for publication.

[13] H.H. Tan, "Tree coding of stationary block ergodic sources with a fidelity criterion", submitted for publication.

[14] I. Csiszar, "On an extremum problem of Information theory", Studia Sci. Math. Hung., Vol. 9, pp. 57-71, 1974.

ENCODING OF A CONTINUOUS SOURCE BY TWISTED MODULATION*

K. YAO
Department of System Science
School of Engineering and Applied Science
University of California
Los Angeles, California 90024

ABSTRACT

The rate-distortion function of a uniformly distributed source under the mean-square error criterion is evaluated by numerical technique. Then the maximum output versus input SNR of an analog communication system with bandwidth expansion for bandlimited i.i.d. uniformly distributed input source is obtained. An upper bound on the mean-square error of the minimum probability of error estimator is found. Then the optimum performance of the system with any twisted modulation can be evaluated. Numerical results indicate tight upper and lower performance bounds at threshold.

I. INTRODUCTION

The efficient transmission of a bandlimited analog message over a channel permitting bandwidth expansion is a problem of basic interest in communication theory. The term twisted modulation has been used to describe such encoding procedure. With rapid increase in the volume of source messages and limited allocation of transmission frequency spectrum, there is considerable practical interest in understanding basic properties of this problem. The problem has been examined originally by Shannon [1]-[2] and Kotelnikov [3]. Recently many others have dealt with this problem [4]-[9]. Despite many known results, basic bounds on performance of analog communication systems are still not known completely.

In this paper, we consider a standard analog communication system model with a bandlimited additive white Gaussian noise. We first obtain the rate-distortion function of a uniformly distributed continuous source under the mean-square error criterion by numerical technique. From this rate-distortion function, the maximum allowable output versus input SNR of any analog communication system with a bandlimited i.i.d. uniformly distributed input source is obtained. These results are parametrized in terms of the bandwidth expansion ratio. Previously known results based on upper and lower bounds of this rate-distortion function leads to an uncertainty of about 1.5dB in output SNR. Secondly, for any analog modulation format, we formulate the optimal demodulator (i.e. estimator) in the minimum probability of estimation error sense. Then we obtain an upper bound on the mean-square estimation error for this optimal demodulation. This result then yields a lower bound on the performance of any analog communication system with this demodulation. By utilizing previously known lower bound on mean-square estimation error and thus upper bound on performance, bounds on optimal performance of the system are obtained. For the well known PFM case, our new bound is seen to be tight. Indeed, with this result, the output SNR at threshold is determined to less than one-half a dB. Previously known uncertainty at threshold is considerably poorer.

---

*This work was supported under ONR Contract No. N00014-69-A-0200-4043, Task No. NR-042-285.

II. ANALOG COMMUNICATION SYSTEM MODEL

We consider a continuous source bandlimited to B Hz. By sampling at the Nyquist rate, a sequence of continuous random variables $\{\lambda\}$ with sampling period $T = 1/2B$, is obtaind. For each λ, the modulator yields a deterministic waveform $S(t,\lambda)$ over each T second interval. The channel is assumed to be an additive white Gaussian noise channel and is bandlimited to W Hz. The noise is assumed to have a zero mean and a two-sided spectral density $N_o/2$. The input of the demodulator is then given by

$$r(t) = s(t,\lambda) + n(t) \ , \quad -T/2 \le t \le T/2 \ . \tag{1.1}$$

The output of the demodulator yields the estimated value $\hat{\lambda}$. The goal of this analog communication system is to produce the sequence of estimates $\{\hat{\lambda}\}$ close to the message sequence $\{\lambda\}$. We use the mean-square error criterion, $\overline{\epsilon^2} = E\{(\hat{\lambda}-\lambda)^2\}$, to characterize the goodness of reproduction for each transmitted λ.

In this paper, we assume $\{\lambda\}$ is a sequence of i.i.d. uniformly distributed random variables on $[-1,1]$. The bandwidth expansion ratio is defined by $\beta = W/\beta$. The output SNR is defined by $\rho_o = \sigma_\lambda^2/\overline{\epsilon^2}$ where the variance of λ for the uniform source is $\sigma_\lambda^2 = 1/3$ and $\overline{\epsilon^2}$ is the mean-square estimation error. The input SNR is defined by $\rho_i = E_o/(N_o/2)$, when $E_o = E\{\int_{-T/2}^{T/2} |S(t,N|^2 dt\} = PT$. The performance of an analog communication system is then characterized by ρ_o versus ρ_i. This performance is then parametrized by β, the modulator, and the demodulator.

III. MAXIMUM OUTPUT VERSUS INPUT SNR

In light of the above defined performance criterion, clearly for each bandwidth expansion ratio β and each input SNR ρ_i, it is desirable to obtain the maximum allowable output SNR ρ_o. Thus, by setting the rate-distortion function $R(D)$ of the source where $D = \overline{\epsilon^2}$ to the channel capacity C, the optimal ρ_o versus ρ_i relationship is obtained. For additive white Gaussian noise channel of bandwidth W Hz, $C = W \log(1+(P/N_oW)) = W \log(1+(2E_o/N_o\beta)) = W \log(1+(\rho_i/\beta))$ bits/sec. From Shannon [1;p. 115], $B \log(Q/D) \le R(D) \le B \log(\sigma_\lambda{}^2/D)$, where Q is the entropy power of the source defined as the power of a Gaussian source which has the same entropy as the message source. Let $K = \sigma_\lambda{}^2/Q$. Upon setting $R(\overline{\epsilon^2}) = C$, then the optimal ρ_o versus ρ_i is given by

$$\rho_L = (1+(\rho_i/\beta))^\beta \le \rho_o \le K(1+(\rho_i/\beta))^\beta = \rho_u \tag{3.1}$$

Since the entropy of the uniform source $H_u = \log 2$ and the entropy of the Gaussian source $H_G = \frac{1}{2} \log 2\pi e Q$, then $K = \pi e/6 = 1.423$. From (3.1) this means the uncertainty of the output SNR ρ_o in dB for any β is

$10[\log \rho_u - \log \rho_L] = 10 \log K = 1.5329\text{dB}$.

From the above discussion, in order to remove this uncertainty in the ρ_o versus ρ_i relationship, it is necessary to obtain the rate-distortion function $R(D)$. Let $R_i(D)$ be the rate-distortion function of a uniformly distributed r.v. on $[-1,1]$ with a mean-sqaure error criterion, then the rate-distortion function of an i.i.d. uniformly distributed source of bandwidth B with a mean-square error criterion is given by $R(D) = 2BR_i(D)$.

It appears this $R_i(D)$ function has not been evaluated previously. For some sources under a given error criterion, the easily evaluable Shannon lower bound actually equals the rate-distortion function at least for some region of distortions. For our source and error criterion, there is strict inequality for all $0 < D < D_{Max}$. However, a necessary and sufficient condition for the solution of the rate-distortion function of a continuous r.v. source with a specified error criterion is known [8 ;p. 475] and [9 ;p. 91]). In practice, the explicit evaluation of the rate-distortion function of a continuous r.v. source by this method can be quite involved [10]. Under the mean-square error criterion, the technique given in[9] requires the solution of a Fredholm integral equation of the first kind on a finite integration range with a Gaussian kernel. Explicit solution of this type of integral equation appears to be difficult.

However, by using the intuition gained in choosing the interior set V_s from [10] and by considering the form of the output probability density $q_s(y)$ in the solution of the $R(D)$ function for a uniform source under an absolute magnitude error criterion [11], we can choose for our problem

$$q_s(y) = \frac{1}{2}\{1+(1-a_s)[\delta(y-a_s) + \delta(y+a_s)]\}, \ y \in V_s \ , \tag{3.2}$$

$q_s(y) = 0$, $y \notin V_s$, and $V_s = [-a_s, a_s]$. Furthermore, for this selected $q_s(y)$, we can use the necessary and sufficient conditions in [9 ,p. 91] based on the $c(y)$ function to determine the validity of the resulting rate-distortion function. Specifically, for our results to be valid, we require

$$\left.\begin{aligned} c(y) &= 1 \ , \quad y \in [-a_s, a_s] \ , && (3.3a) \\ c(y) &\leq 1 \ , \quad y \in [-1,-a_s) \cup (a_s, 1] && (3.3b) \end{aligned}\right\} \quad 0 \leq a_s \leq 1$$

By numerical evaluation, condition (3.3b) is found to be valid for all values of a_s . Condition (3.3a) is satisfied with great accuracy for small to medium values of a_s but loses some accuracy for values of y near $\pm a_s$ as a_s gets near one. Indeed, for a_s approaches one, the integral in $c(y)$ contains delta-like functions and numerical evaluations for (3.3a) is expected to be poor.

Since the expressions for the distortion D and the rate-distortion function $R_i(D)$ depend on $q_s(y)$ in a continuous manner, we can conclude that the numerically evaluated $R_i(D)$ is extremely close to its true value. Results on $R_i(D)$ and related a_s , s , and D are given in Table 1. A plot of $R_i(D)$, $R_L(D) = 0.5\ \ln(2/\pi e D)$, and $R_u(D) = 0.5\ \ln(1/3D)$ are given in Figure 1.

As noted before, by using $2BR_i(D) = R(D) = C$, we obtain

$$R(1/3\ \rho_o) = (\beta/2)\ \log\ (1+(\rho_i/\beta)) \tag{3.4a}$$

$$\rho_o = 1/3\ R_i^{-1}\{(\beta/2)\ \log\ (1+(\rho_i/\beta))\} \tag{3.4b}$$

Since $R_i(\cdot)$ and $R_i^{-1}(\cdot)$ are now available, for any β and ρ_i , the maximum allowable ρ_o can be determined from (3.4b).

IV. BOUNDS ON PERFORMANCE OF ANALOG COMMUNICATION SYSTEMS

Now, consider an analog communication system as discussed in Section II. For any twisted modulation format, the performance of the system clearly depends on the form of the demodulator as well as the parameters in the noise channel. In this Section, we are interested in the performance of the system utilizing an optimal demodulator. In general, since it is not feasible to evaluate the output versus input SNR explicitly, it is necessary to use bounding techniques to obtain performance results. Various lower bounds on parameter estimation mean-square error are known [12]-[14]. These results then yield upper bounds on the performance of the system. Among these bounds, the Cramer-Rao bound [13] is easy to apply and is tight for large ρ_i . However, for ρ_i below threshold, the Cramer-Rao bound is very loose. Recent lower bounds on mean-square error discussed in [14] appears to be best among all known lower bounds. Known results of upper bound on mean-square error for specific demodulator appear to be less developed. The upper bound on mean-square error utilizing ML estimation developed in [15] requires some smoothness conditions on the modulation waveform and also requires numerical integration. At least for the specific PFM case considered in [13], the performance bound obtained using this technique appears to be not useful. In [16], the general technique used in [15] is simplified to computational effort involving only one-dimensional integrations. Nevertheless, it seems the evaluations needed in this approach are still quite involved.

Now, we consider the analog communication system discussed in Section II. Let $P_\lambda(\hat{\lambda}>\lambda_1)$ be the conditional error probability that when λ is transmitted, the estimate $\hat{\lambda}$ satisfies $\hat{\lambda}>\lambda_1$, for any fixed λ_1 . Let $\delta = |\hat{\lambda}-\lambda|$ with range [0,2]. The probability that δ exceeds ϵ , $0 \le \epsilon \le 2$, for uniformly distributed λ on [-1,1] is denoted by

$$G_\delta(\epsilon) = \Pr\{\delta>\epsilon\} = \frac{1}{2}\int_{-1}^{1} [P_\lambda(\hat{\lambda}>\lambda+\epsilon) + P_\lambda(\hat{\lambda}<\lambda-\epsilon)]d_\lambda\ . \tag{4.1}$$

Upon changes in integration variables [3;pp. 90-91], we have

$$G_\delta(\epsilon) = \int_{-1+\epsilon}^{1+\epsilon} \frac{1}{2}P_{\lambda_o-\epsilon}(\hat{\lambda}>\lambda_o)d\lambda_o + \int_{-1-\epsilon}^{1-\epsilon} \frac{1}{2}P_{\lambda_o+\epsilon}(\hat{\lambda}<\lambda_o)d\lambda_o\ . \tag{4.2}$$

For any fixed $0 \le \epsilon \le 2$ and λ uniformly distributed in [-1,1], consider the optimal estimator $\hat{\lambda}_M$ defined by

$$\operatorname*{Inf}_{\hat{\lambda}} \Pr\{|\hat{\lambda}-\lambda| > \epsilon\} = \Pr\{|\hat{\lambda}_M-\lambda| > \epsilon\} = G_\delta^M(\epsilon)\ . \tag{4.3}$$

We also consider any estimator $\hat{\hat{\lambda}}$ operating on a received waveform $r(t)$ of the form given in (1.1) when λ is allowed to be a random variable with range $[-1-2\epsilon,1+2\epsilon]$ for any $0 \le \epsilon \le 2$. Then by using (4.2) and the optimal property of $\hat{\lambda}_M$ in (4.3), we have

$$G_\delta^M(\epsilon) = \int_{-1+\epsilon}^{1+\epsilon} \frac{1}{2} P_{\lambda_o-\epsilon}(\hat{\lambda}_M > \lambda_o)\, d\lambda_o + \int_{-1-\epsilon}^{1-\epsilon} \frac{1}{2} P_{\lambda_o+\epsilon}(\hat{\lambda}_M < \lambda_o)\, d\lambda_o \tag{4.4a}$$

$$\le \int_{-1+\epsilon}^{1+\epsilon} \frac{1}{2} P_{\lambda_o-\epsilon}(\hat{\hat{\lambda}} > \lambda_o)\, d\lambda_o + \int_{-1-\epsilon}^{1-\epsilon} \frac{1}{2} P_{\lambda_o+\epsilon}(\hat{\hat{\lambda}} < \lambda_o)\, d\lambda_o \tag{4.4b}$$

$$\le \int_{-1-\epsilon}^{1+\epsilon} \frac{1}{2}[P_{\lambda_o-\epsilon}(\hat{\hat{\lambda}} > \lambda_o) + P_{\lambda_o+\epsilon}(\hat{\hat{\lambda}} < \lambda_o)]\, d\lambda_o \; . \tag{4.4c}$$

(4.4c) follows from (4.4b) since the integrands are non-negative and $[-1+\epsilon,1+\epsilon] \cup [-1-\epsilon,1-\epsilon] = [-1-\epsilon,1+\epsilon]$. For any fixed $0 \le \epsilon \le 2$ and any fixed $\lambda_o \epsilon [-1-\epsilon,1+\epsilon]$, define the optimal estimator $\hat{\lambda}_M$ by

$$P_{e,M}(\lambda_o,\epsilon) = \operatorname*{Inf}_{\hat{\lambda}} \{\frac{1}{2} P_{\lambda_o-\epsilon}(\hat{\hat{\lambda}} > \lambda_o) + P_{\lambda_o+\epsilon}(\hat{\hat{\lambda}} < \lambda_o)\}$$

$$= \frac{1}{2}\{P_{\lambda_o-\epsilon}(\hat{\hat{\lambda}}_M > \lambda_o) + P_{\lambda_o+\epsilon}(\hat{\hat{\lambda}}_M < \lambda_o)\} \tag{4.5}$$

Now, consider a binary hypothesis testing problem, where under H_o : $r(t) = s(t,\lambda_o-\epsilon) + n(t)$ and H_1: $r(t) = s(t,\lambda_o+\epsilon) + n(t)$, $-T/2 \le t \le T/2$. Let H_o and H_1 be equi-probable. Consider a binary detector consisting of an estimator $\hat{\hat{\lambda}}$ and a threshold decision device that declares H_o if $\hat{\hat{\lambda}} < \lambda_o$ and declares H_1 if $\hat{\hat{\lambda}} > \lambda_o$. Then the probability of a detection error is given by $\frac{1}{2}\{P_{\lambda_o-\epsilon}(\hat{\hat{\lambda}} > \lambda_o) + P_{\lambda_o+\epsilon}(\hat{\hat{\lambda}} < \lambda_o)\}$. From (4.5), it is clear the optimal estimator $\hat{\lambda}_M$ results in the smallest probability of error, $P_{e,M}(\lambda_o,\epsilon)$, for a binary detector in the form of an estimator followed by a single threshold decision device. However, a possible implementation of the well known optimum detector for equi-probable binary deterministic signals in an additive white Gaussian noise channel in the minimum probability of error sense is an estimator, $\hat{\hat{\lambda}} = \int_{-T/2}^{T/2} r(t)[s(t,\lambda_o-\epsilon) - s(t,\lambda_o+\epsilon)]dt+\lambda_o$, followed by a single threshold decision device with threshold set at λ_o [7; Vol. 1, p. 256]. Furthermore, this probability of error is $Q(d/2)$, where

$$Q(x) = (2\pi)^{-1/2} \int_x^\infty \exp(-t^2/2)dt \tag{4.6}$$

$$d^2(\lambda_o,\epsilon) = d^2 = 2 \int_{-T/2}^{T/2} [s(t,\lambda_o-\epsilon) - s(t,\lambda_o+\epsilon)]^2 dt/N_o \qquad (4.7)$$

But this binary detector is in the form of an estimator followed by a single threshold decision device. Thus, $P_{e,M}(\lambda_o,\epsilon) = Q(d/2)$.

In (4.4c), the expression is valid for any $\hat{\hat{\lambda}}$ and in particular for $\hat{\hat{\lambda}}_M$ for any fixed $0 \le \epsilon \le 2$ and any fixed $-1-\epsilon \le \lambda_o \le 1+\epsilon$. Thus, we have

$$G_\delta^M(\epsilon) \le \int_{-1-\epsilon}^{1+\epsilon} Q(d(\lambda_o,\epsilon)/2)d\lambda_o = G_v(\epsilon) \ , \qquad (4.8)$$

when $Q(\cdot)$ and $d(\cdot)$ are defined in (4.6) and (4.7). Now, we can obtain an upper bound on the parameter estimation mean-square error. Let $\overline{\epsilon^2} = E\{\delta^2\} = \int_o^2 \epsilon^2 f_\delta(\epsilon)d\epsilon$, when $F_\delta(\epsilon) = \Pr\{\delta \le \epsilon\} = 1-G_\delta(\epsilon)$, and $f_\delta(\epsilon) = F_\delta'(\epsilon)$. As in [14,p. 340], integration by parts yields

$$\overline{\epsilon^2} = \int_o^2 \epsilon^2 F_\delta'(\epsilon)d\epsilon = 2 \int_o^2 \epsilon G_\delta(\epsilon)d\epsilon \qquad (4.9)$$

The mean-square error $\overline{\epsilon^2}$ is unbounded if we pick an arbitrarily bad estimator $\hat{\delta}$. However, for the optimal estimator $\hat{\lambda}_M$ as defined in (4.3), since $G_\delta^M(\epsilon)$ is bounded, the optimal (i.e., minimum) mean-square error, $\overline{\epsilon_M^2}$, is also bounded. From (4.9) and (4.8), an upper bound to $\overline{\epsilon_M^2}$, denoted by $\overline{\epsilon_v^2}$, is given by

$$\overline{\epsilon_M^2} = 2 \int_o^2 \epsilon G^M(\epsilon)d\epsilon \le 2 \int_o^2 \epsilon G_v(\epsilon)d\epsilon$$

$$= 2 \int_o^2 \epsilon\{ \int_{-1-\epsilon}^{1+\epsilon} Q(d(\lambda_o,\epsilon)/2)d\lambda_o \}d\epsilon = \overline{\epsilon_v^2} \qquad (4.10)$$

Clearly $G_\delta(\epsilon)$ (and in particular $G_\delta^M(\epsilon)$) is a non-increasing function of ϵ . However, $G_v(\epsilon)$ as defined in (4.8) need not be a non-increasing function for $0 \le \epsilon \le 2$. Thus, a tighter (smaller) upper bound on $G_\delta^M(\epsilon)$ and thus $\overline{\epsilon_M^2}$ can be obtained by the technique of leveling $G_v(\epsilon)$. This technique is similar to that used in [3; pp. 94-95] and [14; pp. 340-341], except they level the local "valleys", since their interests were in obtaining tight lower bounds on $G_\delta(\epsilon)$. Here, we level the local "mountain tops" of $G_v(\epsilon)$ and obtain $G_u(\epsilon)$. Thus, $G_\delta^M(\epsilon) \le G_u(\epsilon) \le G_v(\epsilon)$ and replacing $G_v(\epsilon)$ by $G_u(\epsilon)$ in (4.10), we have

$$\overline{\epsilon_M^2} \le \overline{\epsilon_u^2} = 2 \int_o^2 \epsilon G_u(\epsilon)d\epsilon \le \overline{\epsilon_v^2} \ . \qquad (4.11)$$

This is our new upper bound on the mean-square error for the optimal estimator in the minimum probability of estimation error sense. As noted in [3; p. 92], often $d(\lambda_o,\epsilon)$ is not dependent on λ_o. Then the expression for $G_v(\epsilon)$ in (4.8) simplifies to

$$G_v(\epsilon) = 2(1+\epsilon)Q(d(\epsilon)/2) \qquad (4.12)$$

The leveling of $G_v(\epsilon)$ then becomes particularly simple.

Now, consider a Pulse Frequency Modulation (PFM) problem for numerical comparison with previously known performance bounds [13]-[14]. Let $S(t,\lambda) = (2E_o/T)^{1/2}\cos[2\pi(f_o+\lambda F)t]$, $-T/2 \le t \le T/2$, where f_o is a carrier frequency and the time-bandwidth product $\beta_o = 2TF = 40$. Since $W = 2F$ and $B = 1/2T$, then the bandwidth expansion ratio $\beta = 80$. Furthermore, $d^2(\lambda_o,\epsilon) = 2(2E_o/N_o)[1-(\sin 40\pi\epsilon/40\pi\epsilon)]$ is not a function of λ_o. Thus, (4.12) holds and $G_v(\epsilon) = 2(1+\epsilon)Q(\{[1-(\sin 40\pi\epsilon/40\pi\epsilon)]\rho_i/2\}^{1/2})$. The leveling of this $G_v(\epsilon)$ is extremely simple and is essentially not dependent on $\rho_i \in [10\text{dB},20\text{dB}]$. Then $\overline{\epsilon_u^2}$ is evaluated via (4.11) and output SNR ρ_o versus input SNR ρ_i are obtained.

In Figure 2, performance bounds for this PFM communication system problem are plotted. Three upper bounds on performance by Ziv-Zakai [12]-[13], Cramer-Rao [3] and [13], and Bellini-Tartara [14] are labled as curve 1, 2, and 3, respectively. The result in [14] is quite simple to evaluate and appears to yield the tightest known upper bound. Curve 4 is our new lower bound on performance based on the result derived in this paper. Curve 5 is an approximation on performance discussed in [13] and [4]. Curve 6 is the lower bound result of Seidman [13] and [15]. From these results, it is clear the performance of a system using the PFM format and the optimal demodulator in the maximum output versus input SNR sense, lies between curve 3 and curve 4. We note the performance results bounded by these two curves are quite tight. If we define threshold as the point on the performance curve which is 1dB lower than the Cramer-Rao bound straight line, then our threshold occurs at approximately $\rho_i = 16.2$dB. At this point, curve 3 and curve 4 have an uncertainty of less than one-half a dB in ρ_o. The uncertainty between curve 3 and curve 6 is in excess of 20dB in ρ_o. Furthermore, above threshold, our new performance bound approaches the Cramer-Rao bound as expected and desired.

V. CONCLUSIONS

We have considered two basic problems in the evaluation of performance of analog communication systems with bandwidth expansion. First, by obtaining the rate-distortion function of the source under the mean-square error criterion, we found the maximum allowable output versus input SNR for any β over the class of all modulation and demodulation formats. Then for any fixed modulation format, we obtained a new approach to lower bound the performance of the optimal system in the maximum output versus input SNR sense. This new lower bound in conjunction with the best of previously known upper bounds seem to predict the system threshold point quite accurately. Numerical results for a specific modulation format confirm the tightness of this new bound. By having the ability to analyze the system performance accurately and simply for any modulation format, clearly it is of interest to continue on the study of modulations that yield performances approaching the rate-distortion performance curve and their possible implementations.

ACKNOWLEDGEMENT

The author wishes to thank Mr. S. Reisenfeld and Mr. Y. T. Yan for some of the numerical results obtained in this paper.

REFERENCES

1. C. E. Shannon and W. Weaver, The Mathematical Theory of Communication, Urbana, Illinois; University of Illinois Press, 1949.
2. C. E. Shannon, "Communication in the Presence of Noise", Proc. of the IRE, pp. 10-21, January, 1949.
3. V. A. Kotel'nikov, The Theory of Optimum Noise Immunity, New York: McGraw-Hill, 1959.
4. J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering, New York: Wiley, 1965.
5. D. J. Sakrison, Communication Theory: Transmission of Waveforms and Digital Information, New York: J. Wiley, 1968.
6. D. J. Sakrison, Notes on Analog Communication, New York: Van Nostrand-Reinhold, 1970.
7. H. L. Van Trees, Detection, Estimation, and Modulation, Vol. 1-2, New York: J. Wiley, 1968, 1971.
8. R. G. Gallagher, Information Theory and Reiliable Communications, New York: J. Wiley, 1968.
9. T. Berger, Rate-Distortion Theory: A Mathematical Basis for Data Compression, Englewood Cliffs, New Jersey: Prentice-Hall, 1971.
10. H. H. Tan and K. Yao,"Evaluation of Rate-Distortion Functions for a Class of i.i.d. Distributed Sources Under an Ab.-Mag. Criterion," IEEE Trans. on Inf. Th., Vol. IT-21, pp. 59-64, Jan. 1975.
11. K. Yao, "Rate-Distortion Functions with Absolute Value Distortion Measure," Lecture Notes on Source Coding/Ergodic Theory Seminar, Ed. by R. M. Gray, Stanford University, Tech. Rept. No. 6503-2, pp.93-101, Sept. 1974.
12. J. Ziv and M. Zakai, "Some lower bounds on signal parameter estimation", IEEE Trans. Inform. Theory, Vol.IT-15, pp. 386-391, May 1969.
13. L. P. Seidman, "Performance limitations and error calculations for parameter estimation", Proc. IEEE, vol. 58, pp. 644-652, May 1970.
14. S. Bellini and G. Tatara, "Bounds on Error in Signal Parameter Estimation", IEEE Trans on Communication, Vol. COM22, pp. 340-342, March 1974.
15. L. P. Seidman, "An upper bound on average estimation error in nonlinear systems", IEEE Trans. Inform. Theory, Vol. IT-14, pp. 243-250, March 1968.
16. U. Timor, "An Upper Bound on the Estimation Error in the Threshold Region", IEEE Trans. on Inform. Th., Vol. IT-16, pp. 692-698, Nov. 1970.

a_s	S	D_s	$R_i(D_s)$
0	-1.500	.3333	0
.01	-1.500	.3332	.0002
.02	-1.501	.3329	.0006
.05	-1.505	.3309	.0036
.10	-1.522	.3241	.0140
.20	-1.587	.2995	.0525
.25	-1.638	.2831	.0793
.30	-1.702	.2650	.1104
.40	-1.886	.2260	.1832
.50	-2.175	.1872	.2677
.60	-2.646	.1508	.3647
.70	-3.479	.1162	.4834
.75	-4.161	.0988	.5579
.80	-5.184	.0810	.6491
.85	-6.869	.0627	.7668
.90	-10.180	.0436	.9340
.92	-12.646	.0356	1.0279
.94	-16.765	.0273	1.1516
.96	-24.988	.0186	1.3299
.98	-29.482	.0159	1.4056
.995	-29.780	.0158	1.4105

Table I. Values of a_s, s, D_s, and $R_i(D)$ for A Uniform Source with A Mean-Square Error Criterion

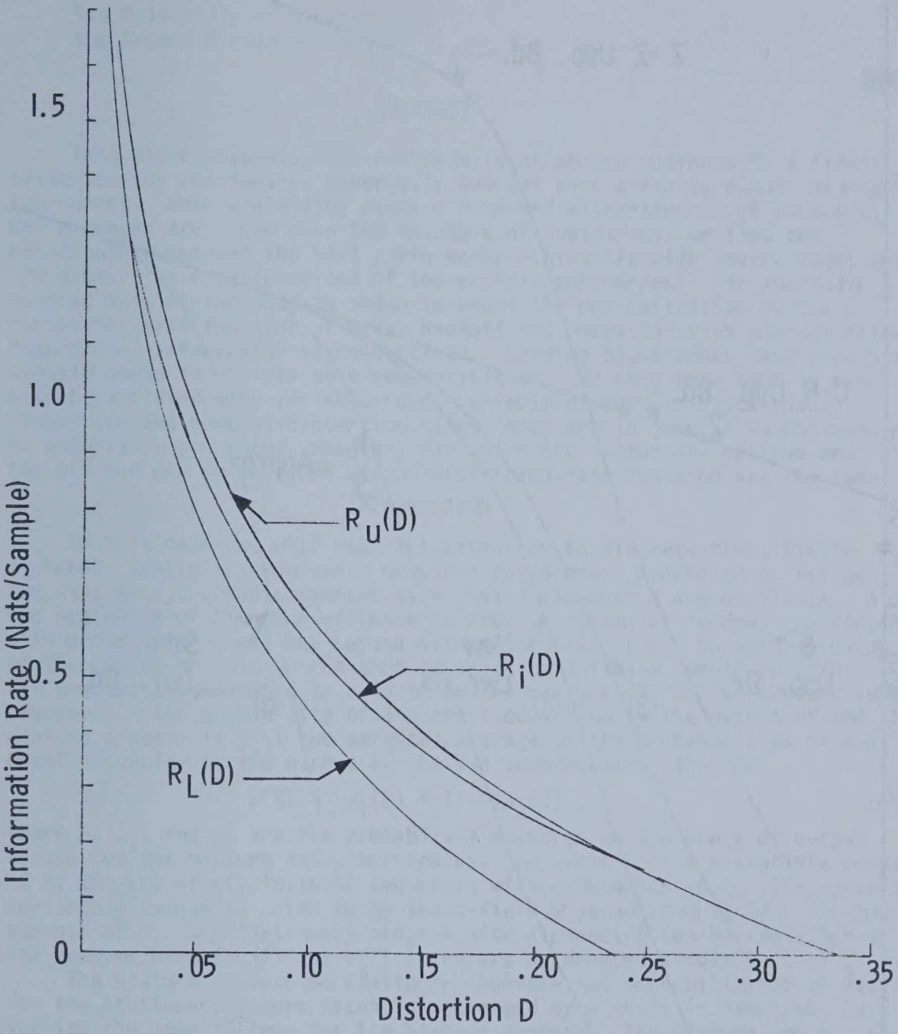


Fig. I. Rate-Distortion Function and Bounds for A Uniform Source with A Mean-Square Error Criterion

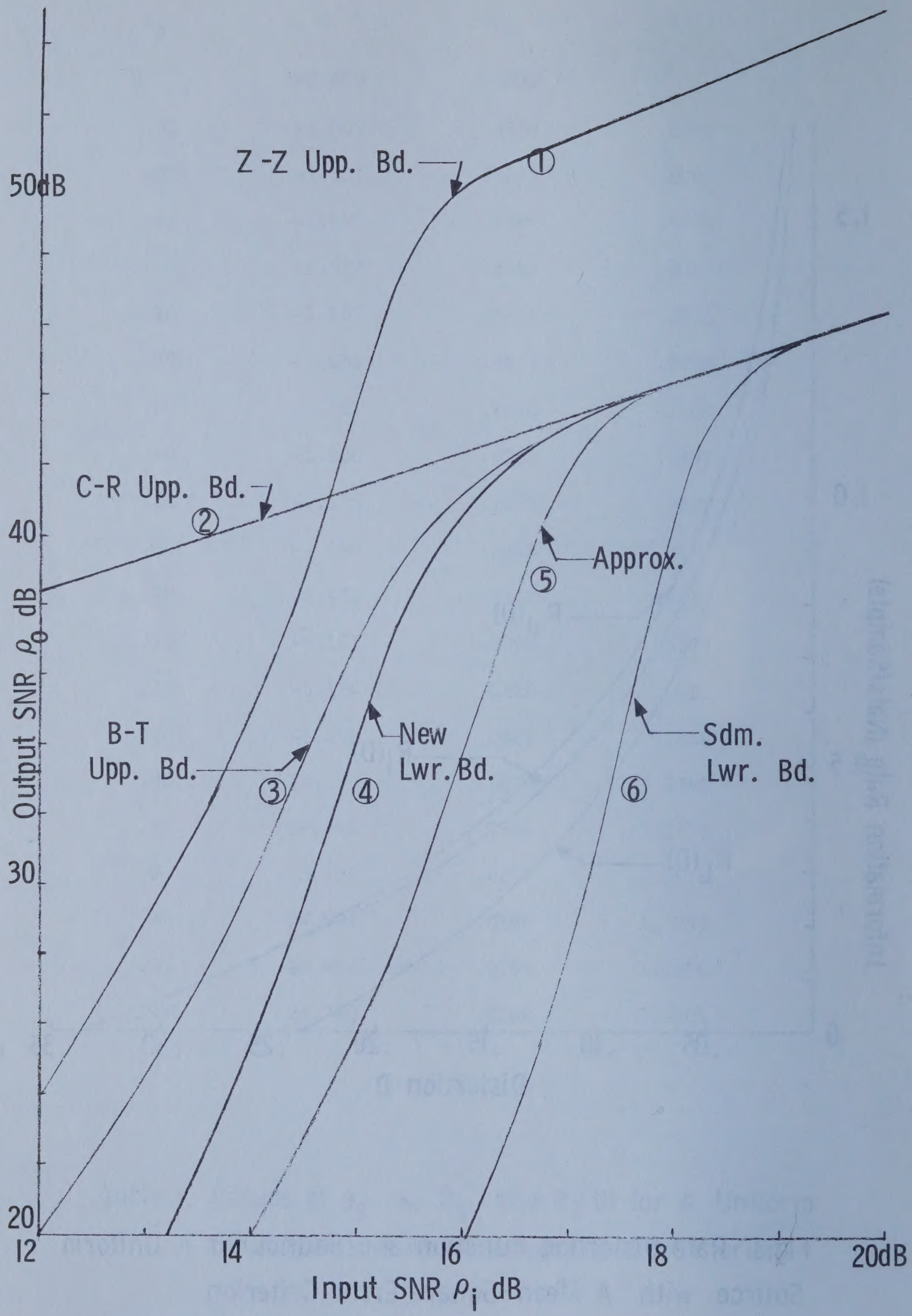


Fig. 2. Output versus Input SNR for PFM with 2TF=40.

OPTIMAL SOURCE CODING FOR STATIONARY NONERGODIC SOURCES

DAVID L. NEUHOFF*
The University of Michigan
Ann Arbor, Michigan 48104

ABSTRACT

This paper presents some new results on source coding with a fidelity criterion for stationary, nonergodic sources with a finite number of ergodic subsources. When minimizing average rate and distortion is of interest, and averages are taken over the ensemble of subsources, we find the relationship between the best performance achievable with source codes and the distortion-rate functions of the ergodic subsources. For aperiodic sources this performance is shown to equal the new definition of the distortion-rate function of Gray, Neuhoff and Omura in which entropy rather than mutual information is constrained. Sliding-block codes have been previously shown to achieve this new definition. We show here that it can also be achieved with variable-rate, variable-distortion block codes. Fixed-rate and bounded-distortion block codes are in general suboptimum. An example is presented, however, for which the latter are optimum and the old and new definitions of the distortion-rate function are the same.

INTRODUCTION

In this paper we will restrict attention to discrete-time, finite-alphabet, stationary sources. Consider the process generated as follows. Two stationary, ergodic sources with finite alphabet A are available. At the beginning of time one of these sources is chosen at random, the first with probability λ and the second with probability $1-\lambda$. Successive outputs of the chosen ergodic source then generate an infinite sample function. The process so generated is said to be the mixture of the two ergodic subprocesses. The probability of any event occurring in the output of the mixture process is just the weighted average of the probabilities of that event occurring in the output of the two subprocesses, that is

$$\mu(E) = \lambda\mu_1(E) + (1-\lambda)\mu_2(E) \ , \tag{1}$$

where μ, μ_1, and μ_2 are the probability measures on the space of output events for the mixture and subprocesses. An event E is a measurable subset of Σ, the set of all infinite sequences with components in A. The space of measurable events is taken to be the σ-field $\mathscr{S}$ generated by the cylinder subsets of Σ. Any stationary process with alphabet A can be specified by its measure μ on $(\Sigma,\mathscr{S})$ and will therefore be denoted simply by μ.

The mixture process is stationary because the probability of an event for the stationary subprocesses is unchanged by a shift in time and Eq.(1) implies the same is true for the mixture process. The mixture process is not ergodic, since the sample functions that occur are representative of one of the ergodic subprocesses but not of the mixture process.

This example is typical of stationary, non-ergodic sources because the ergodic decomposition theorem [1], [2], shows that any stationary source μ with alphabet A can be considered to be the mixture of a collection Λ (possibly infinite) of ergodic subsources in the following sense. The probability of any event is the weighted average of its probabilities under the ergodic measures, i.e.,

---

* Computer, Information, and Control Engineering Program, and the Department of Electrical and Computer Engineering. This research was supported by National Science Foundation Grant ENG 75-10207.

$$\mu(E) = \int_{\Lambda} \mu_{\theta}(E)dW(\theta) \ ,$$

where $\{\mu_\theta\}_{\theta\varepsilon\Lambda}$ are the measures associated with the ergodic subsources. The ergodic decomposition theorem also shows that if f is any bounded measurable function on Σ, then

$$\int f \ dw = \int dw \int f \ d\mu_{\theta} \ . \tag{2}$$

In this paper we will state and prove all results for the simple case just described of a mixture source with two ergodic subsources. All results generalize immediately to any finite mixture of ergodic sources.

If we wish to compress the data from the mixture source and we apply a source code C, which possibly introduces some distortion into the reproduction, then we will find that the time-average rate of compression R in bits per source source symbol and the per letter distortion D will vary depending on which ergodic subsource is actually being encoded. For example, we may get rate R_1 and distortion of D_1 if the first subsource is present, while we may get rate R_2 and distortion D_2 if the second subsource is present, where $(R_1,D_1) \neq (R_2,D_2)$. In this paper we shall take the average rate R and the average distortion D with respect to the mixture distribution as our measures of source coding performance. From Eq.(2) this is simply the average performance over the ensemble of subsources, namely

$$R = \lambda R_1 + (1-\lambda)R_2$$
$$D = \lambda D_1 + (1-\lambda)D_2$$

This measure of performance is of interest, for example, when a number of identical mixture sources are to be source encoded simultaneously and transmitted over a single channel, perhaps with a data concentrator to handle the varying rate from the source encoders. We reiterate that the average performance computed as above will not necessarily be the time-average performance of a code on any particular mixture source.

We will consider the performance of both block and sliding-block source codes on mixture sources.

BLOCK SOURCE CODES

A block code C of blocklength N consists of a codebook, also denoted C, of reproducing words drawn from the N-fold product $\hat{A}^N$ of a reproducing alphabet $\hat{A}$, an approximation rule c: $A^N \rightarrow C$, and a one to one binary encoding rule b: $C \rightarrow B \subset \{0,1\}^*$. $\{0,1\}^*$denotes the set of all finite length binary strings. A source word $x^N = (x_0,\ldots,x_{N-1})\varepsilon A^N$ is encoded for transmission into $W = b(c(x^N)) = b(x^N)$ and then decoded after transmission into $b^{-1}(W) = c(x^N)$. If B is a prefix set, then the code will be uniquely decodable. The average distortion in the decoded reproduction of source μ will be

$$\rho_{\mu}(C) = E_{\mu^N}\rho^N(X^N,c(X^N)) \ ,$$

where $X^N = (X_0,\ldots,X_{N-1})$ are the first N random variables from the source, μ^N is the measure on A^N induced by μ, ρ^N: $\hat{A}^N \times A^N \rightarrow \mathbb{R}$ is the per letter distortion given by

$$\rho^N(x^N,y^N) = \frac{1}{N}\sum_{i=0}^{N-1} \rho(x_i,y_i) \ ,$$

and ρ: $A \times \hat{A} \rightarrow \mathbb{R}$ is a nonnegative single letter distortion measure. Let $\ell(x^N) = \ell(b(c(x^N)))$ denote the length of the binary codeword for x^N. The rate of C is defined to be

$$R_\mu(C) = \frac{1}{N}E_\mu \ell(X^N)$$

There are several ways in which block codes for a mixture source can be constructed from codes with known performance for the ergodic subsources.

Fixed-rate blocks codes

In the most elementary form of a block code the elements of C are ordered, the approximation rule used is

$$c(x^N) = v_i^N \varepsilon C \quad \text{iff } \rho^N(x^N,v_i^N) \leq \rho^N(x^N,v_j^N) \quad \text{all } v_j^N \varepsilon C \ ,$$

and the binary codeword for x^N is simply a binary representation of the index of $c(x^N)$ in C of length $\ell(x^N) = \log\|C\|$. Hence the average rate is

$$R(C) = \frac{1}{N} \log\|C\| \ ,$$

where $\|C\|$ denotes the number of elements in C and the logarithm is to the base two. Such a block code is called fixed-rate since the rate depends only on the properties of the code and not on the data being encoded. We are interested in the best performance, i.e., the least distortion achievable with fixed-rate block codes of rate R, which we denote by $\delta^f(R)$. Gray and Davisson [3] have shown that for a mixture of two ergodic subsources

$$\delta^f(R) = \lambda D_1(R) + (1-\lambda)D_2(R) \ , \tag{3}$$

where $D_i(R)$ is the distortion-rate function of the i^{th} subsource μ_i.

Bounded-distortion block codes

For some D > 0, let C be a blocklength N codebook such that for each $x^N \varepsilon A^N$, there exists $v^N \varepsilon C$ such that $\rho^N(x^N,v^N) \leq D$ and let $b: C \rightarrow B$ be a given binary encoding rule. Let

$$c(x^N) = v_i^N \text{ iff } \rho^N(x^N,v_i^N) \leq D \text{ and } \ell(v_i^N) = \min\{\ell(v_j^N): \rho(x^N,v_j^N) \leq D\}$$

With this approximation rule, C is said to be a bounded-distortion block code since for any source μ

$$\rho_\mu(C) = E_\mu \rho^N(X^N,c(X^N)) \leq D$$

Again we are interested in the best possible performance which can be expressed as the least rate R achievable with bounded-distortion D and which will be denoted $r^b(D)$. From the source coding theorem of Davisson and Pursley [4], it follows that for an ergodic source $r^b(D)$ equals R(D), the rate-distortion function. Note that R(D) is the inverse of D(R). These authors have also shown [5] that for a mixture of two ergodic sources

$$r^b(D) = \lambda R_1(D) + (1-\lambda)R_2(D) \ . \tag{4}$$

Variable-rate, variable-distortion block codes

In general there are block codes with encoding rules such that neither rate nor distortion are fixed, but instead vary upon application to different sources. We now demonstrate what can be done with variable-rate, variable-distortion codes for a mixture source. Let C_1 and C_2 be fixed-rate block codes with blocklengths N, approximation rules c_i, binary encoding rules b_i, rates R_i, and distortions D_i on the subsources μ_i, i=1,2. Let $C = C_1 \cup C_2$ and given $\varepsilon > 0$ let T_1^N, T_2^N be a partition of A^N such that

$$\mu_i^N(T_i^N) \geq 1 - \varepsilon , \quad i=1,2 .$$

We show in the appendix that such a partition exists. Let the approximation rule be

$$c(x^N) = c_i(x^N) \text{ if } x^N \varepsilon T_i^N ,$$

and let the binary encoding rule be given by

$$b(v^N) = \begin{cases} 0 \text{ followed by } b_1(v^N), \text{ if either } v^N \varepsilon C_1 - C_2, \text{ or } v^N \varepsilon C_1 \cap C_2 \text{ and} \\ \qquad\qquad\qquad \ell(b_1(v^N)) \leq \ell(b_2(v^N)) \\ 1 \text{ followed by } b_2(v^N), \text{ if otherwise} \end{cases}$$

The average rate of this code is

$$R(C) = \frac{1}{N} E_\mu \ell(X^N) = \frac{\lambda}{N} E_{\mu_1} \ell(X^N) + \frac{(1-\lambda)}{N} E_{\mu_2} \ell(X^N)$$

Treating the first expectation on the right hand side, we find

$$E_{\mu_1} \ell(X^N) = \int_{T_1^N} \ell(b(x^N)) d_{\mu_1}(x^N) + \int_{T_2^N} \ell(b(x^N)) d_\mu (x^N)$$

$$\leq \int_{T_1^N} (\ell(b_1(x^N))+1) d_{\mu_1}(x^N) + \int_{T_2^N} (\ell(b_2(x^N))+1) d_{\mu_1}(x^N)$$

$$\leq NR_1 + 1 + \varepsilon NR_2 + \varepsilon$$

Similarly, $E_{\mu_2} \ell(X^N) \leq NR_2 + 1 + \varepsilon NR_1 + \varepsilon$, and thus

$$R(C) \leq \lambda R_1 + (1-\lambda) R_2 + \frac{1}{N} + \varepsilon \max\{R_1, R_2\} + \frac{\varepsilon}{N}$$

The resulting average distortion for this code is

$$\rho_\mu(C) = \lambda \rho_{\mu_1}(C) + (1-\lambda) \rho_{\mu_2}(C) ,$$

where we have

$$\rho_{\mu_1}(C) = \int_{T_1^N} \rho^N(x^N, c_1(x^N)) d_{\mu_1}(x^N) + \int_{T_2^N} \rho^N(x^N, c_2(x^N)) d_{\mu_1}(x^N) \leq D_1 + \varepsilon \rho_{max}$$

and similarly $\rho_{\mu_2}(C) \leq D_2 + \varepsilon \rho_{max}$, where $\rho_{max} = \max_{x \varepsilon A, y \varepsilon \hat{A}} \rho(x,y)$. It follows that

$$\rho_\mu(C) \leq \lambda D_1 + (1-\lambda) D_2 + \varepsilon \rho_{max} .$$

This construction has been proposed independently by L. Davisson, M. Pursley, and K. Mackenthun. As usual we are interested in the best performance achievable with variable-rate, variable-distortion block codes, which we can express either as $\delta(R)$, the least distortion achievable with such codes of rate R, or the inverse of $\delta(R)$, namely $r(D)$, the least rate achievable with codes giving distortion D. Since in the above construction we can make N arbitrarily large, ε arbitrarily small, and C_1 and C_2 arbitrarily close to optimum for the two subsources, i.e., $D_1 \approx D_1(R_1)$ and $D_2 \approx D_2(R_2)$, we can see that if $R = \lambda R_1 + (1-\lambda)R_2$, then

$$\delta(R) \le \lambda D_1(R_1) + (1-\lambda)D_2(R_2) .$$

Taking the over all R_1,R_2 satisfying $\lambda R_1 + (1-\lambda)R_2 \le R$ we find

$$\delta(R) \le \inf_{R_1,R_2:\ \lambda R_1 + (1-\lambda)R_2 \le R} \lambda D_1(R_1) + (1-\lambda)D_2(R_2)$$

This result could also have been established by combining two bounded-distortion codes. The next theorem establishes the converse and restates the results.

Theorem 1.

(i) $$\delta(R) = \inf_{R_1,R_2:\ \lambda R_1 + (1-\lambda)R_2 \le R} \lambda D_1(R_1) + (1-\lambda)D_2(R_2) \tag{5a}$$

(ii) $$r(D) = \inf_{D_1,D_2:\ \lambda D_1 + (1-\lambda)D_2 \le D} \lambda R_1(D_1) + (1-\lambda)R_2(D_2) \tag{5b}$$

(iii) $D(R) \le \delta(R)$

(iv) $\delta(R) \le \delta^f(R)$, $\delta(R) \le \delta^b(R)$.

Proof. We will prove (i), the proof for (ii) follows since the inverses of the left and right hand side of (i) are the left and right hand side of (ii). We have already established half the result. It remains to show that

$$\delta(R) \ge \inf_{R_1,R_2:\ \lambda R_1 + (1-\lambda)R_2 \le R} \lambda D_1(R_1) + (1-\lambda)D_2(R_2) .$$

Let C be any variable-rate, variable-distortion block code with blocklength N, encoding function b and approximation rule c such that $\rho_\mu(C)=D$ and $R_\mu(C)=R$.

$$\begin{aligned} \rho_\mu(C) &= \lambda\rho_{\mu_1}(C) + (1-\lambda)\rho_{\mu_2}(C) \\ &\ge \lambda D_1(R_{\mu_1}(C)) + (1-\lambda)D_2(R_{\mu_2}(C)) \\ &\ge \inf_{R_1,R_2:\ \lambda R_1 + (1-\lambda)R_2 \le R} \lambda D_1(R_1) + (1-\lambda)D_2(R_2) \end{aligned} \tag{6}$$

The first inequality follows from the converse to the source coding theorem for ergodic sources, and the second inequality follows from the fact that $\lambda R_{\mu_1}(C) + (1-\lambda)R_{\mu_2}(C) = R_\mu(C) = R$. Since Eq.(6) is true for all codes C, the result follows. (iii) $D(R) \le \delta(R)$ follows from the general converse to the source coding theorem [6]. (iv) $\delta(R) \le \delta^f(R)$ and $\delta(R) \le \delta^b(R)$ follow from the fact that fixed-rate and bounded-distortion codes are merely special cases of variable-rate, variable distortion codes.

Since $\delta(R)$ is the infimum of a convex function over a convex set the infimum must be attained by some (R_1^*, R_2^*). The following theorem provides a way of parametrically generating $\delta(R)$ or $r(D)$.

Theorem 2. If $\frac{d}{dR} D_i(R)$ exists and is continuous for all $R \geq 0$ and $i = 1,2$, then necessary and sufficient conditions for R_1^*, R_2^* to achieve the infimum in (i) of Theorem 1 are (a) $\lambda R_1^* + (1-\lambda) R_2^* = R$ and (b) there exists α such that

$$\frac{d}{dR} D_i(R) \Big|_{R=R_i^*} \geq \alpha \qquad i = 1,2$$

with equality iff $R_i^* > 0$.

This theorem follows from Theorem 4.4.1 of Gallager [7, p.87]. It shows that a point of $\delta(R)$ (or equivalently $r(D)$) can be found by averaging $D_1(R)$ and $D_2(R)$ at points of equal slope. If $\frac{d}{dR} D_1(R) \Big|_{R_1} = \frac{d}{dR} D_2(R) \Big|_{R_2}$, then $\delta(\lambda R_1 + (1-\lambda) R_2) = \lambda D_1(R_1) + (1-\lambda) D_2(R_2)$. In general, the slope of a distortion-rate function increases monotonically from some negative minimum value at $R = 0$ to zero at R equal to the entropy-rate of the source. If the i^{th} source has smaller slopes than the j^{th}, then there will be points of $\delta(R)$ obtained by averaging $D_i(R_i)$ at $R_i > 0$ with $D_j(R_j)$ at $R_j = 0$ resulting in $\delta(\lambda_i R_i) = \lambda_i D_i(R_i) + \lambda_j D_j(0)$ or equivalently $r(\lambda_i D_i + \lambda_j D_j) = \lambda_i R_i(D_i)$.

An example.

Let μ be the mixture of two binary coin flipping processes with independent tosses and with probability of HEAD p in the first process μ_1 and q in the second process μ_2, $p < q \leq \frac{1}{2}$. Let the mixture distribution give equal probability to each subsource, i.e., $\mu = \frac{1}{2}\mu_1 + \frac{1}{2}\mu_2$. We will use Theorem 2 to compute $r(D)$. The rate-distortion functions of the subsources are [8, p.47]

$$R_1(D) = \begin{cases} h(p) - \log D & D \leq p \\ 0 & D \geq p \end{cases}$$

and

$$R_2(D) = \begin{cases} h(q) - \log D & D \leq q \\ 0 & D \geq q \end{cases},$$

where $h(p) = -p \log p - (1-p) \log (1-p)$. We notice that $R_1(D)$ and $R_2(D)$ have the same slopes for all $D \leq p$, and hence

$$r(D) = \tfrac{1}{2}R_1(D_1) + \tfrac{1}{2}R_2(D_2) = \tfrac{1}{2}h(p) + \tfrac{1}{2}h(q) - h(D), \quad D \leq p .$$

The remainder of $r(D)$ can be found by averaging $R_1(D_1)$ at $D_1 = p$ where $R_1(p) = 0$ with $R_2(D_2)$ for $D_2 \geq p$. This gives

$$r(D) = \tfrac{1}{2}R_2(2D-p) = \tfrac{1}{2}h(q) - \tfrac{1}{2}h(2D-p), \quad D \geq p.$$

We plot $R_1(D)$, $R_2(D)$, $r^b(D) = \frac{1}{2}R_1(D) + \frac{1}{2}R_2(D)$, and $r(D)$ in Figure 1. Notice that when $D \leq p$, $r(D) = r^b(D)$. This means that bounded-distortion codes are optimum in this range. We have also plotted $r^f(D)$, the best performance achievable with fixed-rate codes. $r^f(D)$ is the inverse of $\delta^f(D)$, which is given by Eq.(3). We observe that $r^f(D)$ is strictly greater than $r(D)$ except at $D = \frac{p+q}{2}$. Further, we notice that $r^b(D) < r^f(D)$ in the region of low distortion, whereas $r^f(D) < r^b(D)$ in the region of high distortion.

SLIDING-BLOCK SOURCE CODES

A sliding-block source code $f^{(N)}$ of blocklength 2N+1 consists of an approximation function $f^{(N)}: A^{2N+1} \to \hat{A}$ and a one-to-one binary encoding rule $b^{(M)}: \hat{A}^M \to B \subset \{0,1\}^*$. Source data is encoded as follows. First the approximation function generates a reproduction of x_k, $\hat{x}_k = f^{(N)}(x_k^{(2N+1)})$, where $x_k^{(2N+1)} = (x_{k-N},\ldots,x_k,\ldots,x_{k+N})$. Then the reproduction sequence $\underline{\hat{x}} = \ldots\hat{x}_{-1}\hat{x}_0\hat{x}_1\hat{x}_2\ldots$ is encoded in blocks of M into the binary codeword $b^{(M)}(\hat{x}^M)$. If B is a prefix set, the uncoded binary data can be correctly decoded into the reproduction sequence $\underline{\hat{x}}$. The average rate of this code is

$$R_\mu(f^{(N)}) = E\,\frac{1}{M}\ell(b^{(M)}(f^{(N)}(X_0^{(2N+1)}),\ldots,f^{(N)}(X_{M-1}^{(2N+1)})))$$

and the average distortion is

$$\rho_\mu(f^{(N)}) = E_\mu \rho(X_0, f^{(N)}(X_0^{(2N+1)}))$$

The reproduction process $\hat{\mu} = \{\hat{X}_k\}$ generated by the approximation rule can only have less entropy-rate than the original source. If optimally designed, the binary encoding rule will compress the reproduction sequence into $H(\hat{X})$ bits per symbol, where $H(\hat{X})$ is the entropy-rate of $\hat{\mu}$. Notice that a sliding-block code is a variable-rate, variable-distortion source code since both rate and distortion depend on the source being encoded.

Let $\delta^S(R)$ denote the least average distortion achievable with sliding-block codes of average rate R for a particular source. The sliding-block source coding theorem [9], [6] in conjunction with the variable-length noiseless source coding theorem [7, p. 58] shows that for a stationary, aperiodic source,

$$\delta^S(R) = D'(R)\ , \tag{7}$$

where

$$D'(R) \triangleq \lim_{N\to\infty} \frac{1}{N} \inf_{p^N \in \mathscr{P}_\mu^N(R)} E_{\mu^N p^N} \rho^N(X^N,\hat{X}^N)\ .$$

$\mathscr{P}_\mu^N(R)$ is the set of all test channels $p^N(\hat{x}^N|x^N)$ such that

$$H(\hat{X}^N) = -\sum_{\hat{x}^N \in \hat{A}^N} q(\hat{x}^N)\log q(\hat{x}^N) \le R\ ,$$

where
$$q(\hat{x}^N) = \sum_{x^N} \mu^N(x^N)p^N(\hat{x}^N|x^N)\ . \tag{8}$$

A source is aperiodic if it assigns zero probability to the set of all periodic sequences in $\sum$. For any stationary source $D'(R) \ge D(R)$ [9], [6].

The next theorem relates the best performance achievable by sliding-block codes as given by Eq.(7) to that of block codes.

Theorem 3. For a stationary, aperiodic mixture source, i.e., one with no periodic, ergodic subsources,

$$\delta^S(R) = \delta(R) .$$

Proof.

(i) The general converse to the source coding theorem [9], [6], shows that for any stationary source, $\delta(R) \geq D'(R) = \delta^S(R)$.

(ii) Using the same procedure used to prove Theorem 1, part (i), it can be established that

$$\delta^S(R) \geq \inf_{R_1,R_2:\ \lambda R_1 + (1-\lambda)R_2 \leq R} \lambda D_1(R_1) + (1-\lambda)D_2(R_2) = \delta(R) .$$

THE DISTORTION-RATE FUNCTION D(R)

Finally we wish to relate $\delta(R)$ to Shannon's original definition of the distortion-rate function

$$D(R) \triangleq \lim_{N\to\infty} \inf_{p^N \varepsilon \mathscr{Q}_\mu^N(R)} \frac{1}{N} E_{p^N} \rho^N(X^N,\hat{X}^N) ,$$

where $\mathscr{Q}_\mu^N(R)$ is the set of all test channels $p^N(\hat{x}^N|x^N)$ such that

$$I_{p^N}(X^N,\hat{X}^N) = \sum_{\substack{x^N \varepsilon A^N \\ \hat{x}^N \varepsilon \hat{A}^N}} \mu^N(x^N)p^N(\hat{x}^N|x^N)\log \frac{p^N(\hat{x}^N|x^N)}{q(\hat{x}^N)} \leq R$$

and $q(\hat{x}^N)$ is given by Eq.(8). In this definition mutual information rather than entropy is constrained.

In the next theorem we employ the Wyner-Ziv lower bound to the rate-distortion function [10] to give sufficient conditions for D(R) to equal $\delta(R)$, when the distortion measure is Hamming distance, $\rho(x,y) = 0$, if $x=y$ and $\rho(x,y) = 1$, if $x \neq y$. In this case the Wyner-Ziv lower bound is (see [8, p.131] and [7, p.407]).

$$R(D) \geq R^{WZ}(D) = H(X) - h(D) - D \log(\|A\|-1) .$$

Theorem 4. If for some rate R, the Wyner-Ziv lower bound to the rate-distortion function holds with equality for each of the ergodic subsources, then

$$D(R) = \delta(R)$$

Proof. By hypothesis

$$\begin{aligned} r(D) \leq \lambda R_1(D)+(1-\lambda)R_2(D) &= \lambda R_1^{WZ}(D)+(1-\lambda)R_2^{WZ}(D) \\ &= \lambda H_1(X)-\lambda h(D)-\lambda D\log(\|A\|-1)+(1-\lambda)H_2(X)-(1-\lambda)h(D) \\ &\quad -(1-\lambda)D\log(\|A\|-1) \end{aligned}$$

$$= H(X)-h(D)-D(\log(\|A\|-1) =$$

$$= R^{WZ}(D) \le R(D) .$$

We have used the fact that the entropy-rate of μ , $H(X)$, equals $\lambda H_1(X)+(1-\lambda)H_2(X)$, [11]. Since we already know $r(D) \ge R(D)$, the desired result follows.

We have also proved the following corollary.

Corollary. Under the same hypotheses as Theorem 4,

$$r^b(D) = R(D) .$$

In the previously discussed example the Wyner-Ziv lower bound is tight for both subsources if $0 \le D \le p$. Hence $R(D) = r(D)$ in this region. It is not known whether or not $R(D) = r(D)$ in the region $p \le D \le \frac{p+q}{2}$ except that $R(\frac{p+q}{2}) = r(\frac{p+q}{2}) = 0$.

Appendix: If $\mu_1 \ne \mu_2$, then the ergodic theorem implies there exists a partition of $\sum$ into measurable sets $T_1, T_2 \varepsilon \mathscr{S}$ such that $\mu_i(T_j) = 0$, $i \ne j$. Since $\mathscr{S}$ is generated by the cylinder sets, T_1 and T_2 can be approximated arbitrarily well with cylinder sets. In particular, given $\varepsilon>0$ there must exist N sufficiently large and cylinder sets $\tilde{T}_i^N = \{x: x_k \varepsilon G_k^i, 0 \le k \le N-1\}$, i=1,2 such that $\tilde{T}_i^N \cap \tilde{T}_2^N = \phi$ and $\mu_i(\tilde{T}_j^N \Delta T_j) < \varepsilon$, where $E \Delta F = (E-F) \cup (F-E)$. This implies

$$\mu_i(\tilde{T}_j^N) = \mu_i(T_i \cap \tilde{T}_j^N) \le \mu_i((T_i \cap T_j) \cup (T_j \Delta \tilde{T}_j^N))$$

$$\le \mu_i(T_i \cap T_j) + \mu_i(T_j \cap \tilde{T}_j^N) \le \varepsilon .$$

We then let $T_i^N = \mathop{X}\limits_{k=0}^{N-1} G_k^i$, i=1,2 . It follows that T_1^N, T_2^N form a partition of A^N such that $\mu_i^N(T_j^N) \le \varepsilon$, $i \ne 1$.

REFERENCES

[1] V. A. Rochlin, "On the fundamental ideas of measure theory," Amer. Math. Soc. Translation, No. 71, 1952.

[2] R. M. Gray and L. D. Davisson, "The ergodic decomposition of discrete stationary sources," IEEE Trans. Inform. Theory, Vol. IT-20, Sept. 1974, pp. 625-636.

[3] R. M. Gray and L. D. Davisson, "Source coding without the ergodic assumption," IEEE Trans. Inform. Theory, Vol. IT-20, July 1974, pp. 502-516.

[4] L. D. Davisson and M. B. Pursley, "A direct proof of the coding theorem for discrete sources with memory," IEEE Trans. Inform. Theory, Vol. IT-21, May 1975, pp. 310-317.

[5] M. B. Pursley and L. D. Davisson, "Variable rate coding for nonergodic sources and classes of ergodic sources subject to a fidelity criterion," submitted to IEEE Trans. Inform. Theory.

[6] R. M. Gray, D. L. Neuhoff, and J. K. Omura, "Process definitions of distortion-rate functions and source coding theorems," IEEE Trans. Inform. Theory, Vol. IT-21, Sept. 1975, pp. 524-532.

[7] R. G. Gallager, Information Theory and Reliable Communication. New York: Wiley, 1968.

[8] T. Berger, Rate Distortion Theory. Englewood Cliffs, N.J.: Prentice Hall, 1971.

[9] D. L. Neuhoff, "Source coding and distance measures on random processes," Ph.D. dissertation, Stanford University, 1974.

[10] A. Wyner and J. Ziv, "Bounds on the rate-distortion function for stationary sources with memory," IEEE Trans. Inform. Theory, Vol. IT-17, 1971, pp. 508-513.

[11] L. Breiman, "On achieving channel capacity in finite memory channels," Ill. J. Math., Vol. 4, pp. 246-252.

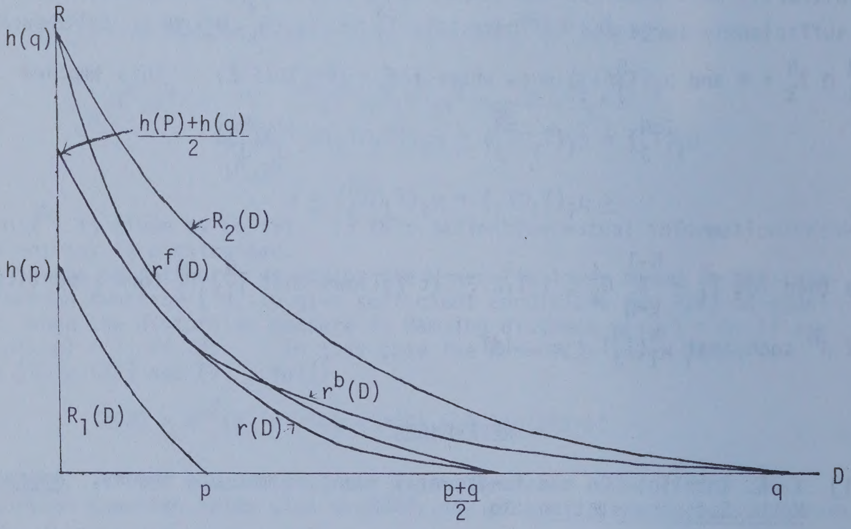


Figure 1.

IMAGE ENCODING WITH A FIDELITY CRITERION

BARRY M. LEINER
Georgia Institute of Technology
Atlanta, Georgia 30332

ABSTRACT

The problem of compressing pictorial data for transmission is investigated. It is seen that the classic measures of distortion such as mean-squared error are not indicative of the quality of the picture. A new measure is suggested and some of the theoretical properties related to the optimum coding (i.e., rate-distortion function) are considered.

INTRODUCTION

In transmitting images, the capacity of the channel is often not sufficient to allow perfect reproduction of the image at the receiver without taking an inordinately long time for transmission. One solution to this problem is to encode the images to remove any redundancy, allowing accurate reproduction of the image at a lower bit rate. This bit stream then requires a lower capacity channel or alternatively, less time to transmit each image. The capacity may still, however, not be adequate.

Requiring perfect reproduction is often unrealistic. In a black and white picture, for instance, relative intensity of each point is often more important than the exact intensity. When the statistics of the image are known as well as a quantitative fidelity measure, the minimum number of bits required to reproduce the image within a given fidelity may be calculated using the techniques of rate-distortion theory. We can then try to find various encoding schemes which would hopefully come close to this lower bound.

Unfortunately, the application of rate-distortion theory requires knowledge of both the source (image) statistics and an adequate fidelity measure. While an estimate of the statistics of the image may be obtained using histogram techniques on a large sample of images of the type concerned with, determining the appropriate fidelity measure is a more difficult problem. Classically, the fidelity measure used is a mean-squared error criterion. Although this measure is analytically "nice," it does not correlate well with human perception of the characteristics for a good reproduction.

In this paper, fidelity criteria are discussed for images that are computationally tractable. The distortion measure chosen is motivated in part by recent psychophysical tests and in part by the desire for a tractable measure for images. It is seen that, while this measure leads to a well-defined and theoretically computable distortion-rate function, in practice its computation is difficult. The remainder of the paper deals with the development of bounds to this distortion-rate function, thereby

---

This work was partially supported by the National Science Foundation under Grant No. ENG75-04992.

developing bounds to the performance of any data compression technique with the chosen fidelity criterion.

DATA COMPRESSION FOR IMAGES

The problem of the transmission of images may be stated as follows. A sender has an image which he would like reproduced for the receiver. The sender has available to him a communication link which operates at some rate. If we allow the sender and receiver to install coding and decoding equipment, what is the best reproduction of the image available to the receiver after some fixed amount of time?

Figure 1 shows the type of system envisioned. The image is first converted to an electrical signal. This would typically be accomplished by a scanning device whose output depends on the pointwise intensity of the image. At the converter output, we then have a signal, U, that completely represents the image. This signal has a data rate which is too high for the channel and must be encoded, giving the signal U_1. To transmit this signal over the channel reliably, it must also be encoded giving the signal U_2 which is sent over the channel. The received signal is then decoded to give $\hat{U}$, an electrical representation of the image. This representation can then be converted to a reproduction of the image. If the channel capacity is not sufficient to transmit a perfect reproduction of the image, the receiver will view a distorted version of the original image. Before proceeding, it should be noted that, in general, the image encoder and channel encoder (and likewise for the decoders) should be shown as a single block. The fact that, under a wide variety of circumstances including the present, the source (image) encoding may be done independent of the channel encoding is a major result of Information Theory [1, p. 3]. Since the interest here is in the required data rate for image reproduction, we shall not be concerned with the channel coding. It will be assumed that the channel has some fixed maximum data rate and never makes any errors.

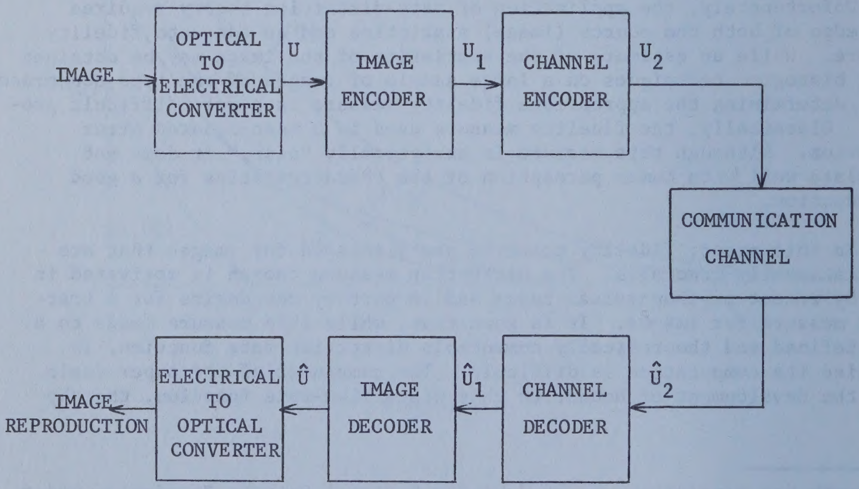


Figure 1. Image Transmission Systems

The system to be analyzed may be redrawn as shown in Figure 2. The representation of the image U is encoded and decoded into a reproduction $\hat{U}$. We desire that the distortion between U and $\hat{U}$ be as small as possible given that the data rate of the encoded image is less than the maximum data rate of the channel.

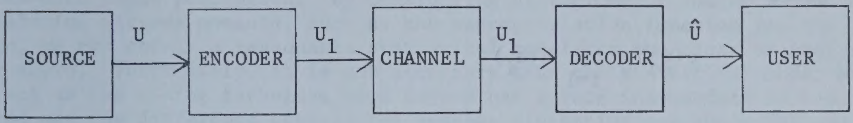


Figure 2. Image Coding System

This may be stated precisely as follows. We are given a set of possible images, S, along with a probability measure, q, for the images. The source produces images from the set S at some rate, say one per second. Each image is produced independently of the other images according to the given probability measure. The set of images may be all possible pages of text, all possible pictures, etc. The probability measure may be allowed to determine the likely images. We may, therefore, take S to be the set of all possible intensity patterns on the square $[0, L] \times [0, L]$. We are envisioning a source which produces a set of uncorrelated images that need to be transmitted at some rate. It should be noted that, by allowing the images to be correlated, we may consider the problem of television transmission in the same framework.

We are also given a set of possible reproduction images $\hat{S}$ which are available at the receiver, along with a measure of the distortion between source and reproduction $d(U, \hat{U}) \geq 0$ defined on $S \times \hat{S}$. We desire to reproduce the images at the receiver with as small an average distortion as possible while keeping the data rate below R. To accomplish this, we will use a block coding technique. Let $\underset{\sim}{U}_n = (U_1, U_2, \ldots, U_n)$, i.e., a set of n images, and similarly for $\hat{\underset{\sim}{U}}_n$. Define the distortion between the image n-tuples as just the average distortion, i.e.,

$$d(\underset{\sim}{U}_n, \hat{\underset{\sim}{U}}_n) = n^{-1} \sum_{i=1}^{n} d(U_i, \hat{U}_i) \quad .$$

A code of block length n, C_n, is simply a collection of a finite number of reproduction image n-tuples. The size of the code $|C_n|$ is just the number of code words (n-tuples) in C_n. The encoding technique is simply to choose the code word in C_n that has minimum distortion with the source word, i.e., $\underset{\sim}{u}_n$ is encoded into $\hat{\underset{\sim}{u}}_n(\underset{\sim}{u}_n)$ if

$$d(\underset{\sim}{u}_n, \hat{\underset{\sim}{u}}_n(\underset{\sim}{u}_n)) = \min_{\hat{\underset{\sim}{u}}_n \in C_n} d(\underset{\sim}{u}_n, \hat{\underset{\sim}{u}}_n) \quad .$$

The code, therefore, has an average distortion

$$d(C_n) = E \min_{\underset{\sim}{\hat{u}}_n \in C_n} d(\underset{\sim}{U}_n, \underset{\sim}{\hat{u}}_n)$$

where the expectation is taken with respect to the probability measure on S^n given by

$$q(\underset{\sim}{u}_n) = \prod_{i=1}^{n} q(u_i) \quad .$$

We note that, if the channel is capable of transmitting R bits per second, then the encoded version of the images can be transmitted over the channel as long as the rate of the code, $R(C_n) = n^{-1} \log |C_n|$, is less than R. The minimum distortion at rate R for codes of block length n is then given by choosing the best code so that

$$d_n(R) = \min_{C_n : R(C_n) \le R} d(C_n) \quad .$$

Since we are not restricted to any particular length code, the minimum distortion achievable at rate R is just $d(R) = \inf_n d_n(R)$.

This minimum distortion is usually impossible to calculate directly as the class of codes is large and the computations very difficult [2]. Shannon [3] showed that d(R) may be related to a well-defined information theoretic minimization in certain circumstances which is amenable to fast computer programming techniques [4]. This famous result, the coding theorem for sources subject to a fidelity criterion, has been generalized several times [1,5,6] and may be stated as follows for memoryless sources (the generalization to sources with correlated images may be found in the references). Let $p(\hat{u}|u)$ be a conditional probability measure (test channel) of reproduction images given sources images. This test channel may be thought of as a random coding device. Let I(q, p) be the average mutual information between U and $\hat{U}$ [1] under the joint probability measure induced by q and p and let D(q, p) be the average distortion, i.e., $D(q, p) = E\, d(U, \hat{U})$. The distortion-rate function for the source with alphabet S, probability measure q, and fidelity measure d is then defined by

$$\delta(R) = \inf_{p : I(q, p) \le R} D(q, p) \quad .$$

Thus, δ(R) is the minimum distortion under a random encoding rule with rate replaced by mutual information while d(R) is the minimum distortion under a deterministic code. The source coding theorem provides the meaning of the distortion-rate function [1,3,5].

Source Coding Theorem: Let U be a discrete time memoryless source as defined above and assume there exists a finite set of reproduction images $\{\hat{U}_j\}$ such that $E \min_j d(U, \hat{U}_j) < \infty$. Then $d(R) = \delta(R)$.

The source coding theorem therefore guarantees us that, as long as there is any finite code having finite average distortion, we can for long code blocklengths, come close to the distortion-rate function. Thus, the distortion-rate function provides a tight lower bound on the average distortion which is achievable by any real coding system.

Unfortunately, to apply the theory outlined above, it is necessary to have both a probability measure for the source and a fidelity measure which agrees with human perception. By considering histograms of images and calculating various moments, such as the autocorrelation function for the image, we can obtain a reasonable statistical model for the class of images of concern. Fortunately, it is not necessary that the statistical model be perfect as the coding technique once chosen has a rate independent of the source and the difference between the average distortion for the model and the real source may be bounded [24]. Alternately, the distortion for a class of sources having the given statistics may be considered [23]. Thus, an analysis based just on a model derived from first and second order statistics will provide a great deal of useful information.

FIDELITY CRITERIA FOR IMAGES

Finding a meaningful fidelity measure is a more difficult problem. Much of the previous work on image data compression has used either a mean-squared error criterion [7,8,9,10,11] or a frequency-weighted mean-squared error [12,16]. While it was recognized that mean-squared error is a poor criterion for images [13,14], its simplicity of calculation plus a lack of a better criterion has made it popular.

The search for a better criterion must be carried out by turning to models of the visual system. Recent psychophysical testing [20,21,22,25] seems to indicate that the eye detects information by passing the scene through a set of spatial channels and then performs an "or" operation on the output of the channels. There is also evidence [13,16] that prior to channelizing, a pointwise nonlinear operation of the form $f(x) = x^{.3}$ is performed. A model of the visual detection system that would include these results in shown in Figure 3. Given this model, it is easily seen why a mean-squared criterion, weighted or not, would not correlate with a human perception of distortion. A given level of mean-squared error could be achieved by having a low level of distortion for each channel or a high level of distortion for one channel. The first may not be detected by any channel and would therefore be perceived as a good reproduction while the second would be detected on the one channel and be perceived as a poor reproduction.

What therefore is desired is a criterion that, when its value is low, guarantees that the distortion on each channel is low. Taking the maximum distortion over the given channels would achieve this goal. A class of candidate distortion measures for images is therefore the following. Let $U(x,y)$ be the intensity pattern of the original image and let $\hat{U}(x,y)$ be the intensity pattern of the reproduction image, where x and y vary from 0 to L, the size of the image. A nonlinear point by point transformation is first performed on both the image and its reproduction, giving $W(x,y) = f(U(x,y))$ and $\hat{W}(x,y) = f(\hat{U}(x,y))$. The transformed images are then passed through a set of ideal two-dimensional bandpass filters, $\{H_i(f_x, f_y)\}_{i=1}^{N}$, giving outputs $V_i(x,y)$ and $\hat{V}_i(x,y)$ for each filter. These filters are chosen so as to completely cover the spatial frequencies up to the human resolution of 40 cycles/deg [13]. The squared error in each channel,

$$d(V_i, \hat{V}_i) = \int_0^L \int_0^L (V_i(x,y) - \hat{V}_i(x,y))^2 \, dxdy$$

is then calculated. The distortion between the original image and its reproduction is then given by $d(U, \hat{U}) = \sup_i (a_i \, d\, (V_i, \hat{V}_i))$ where the a_i are positive constants. Work is currently underway by the author to verify this distortion measure and determine its associated parameters.

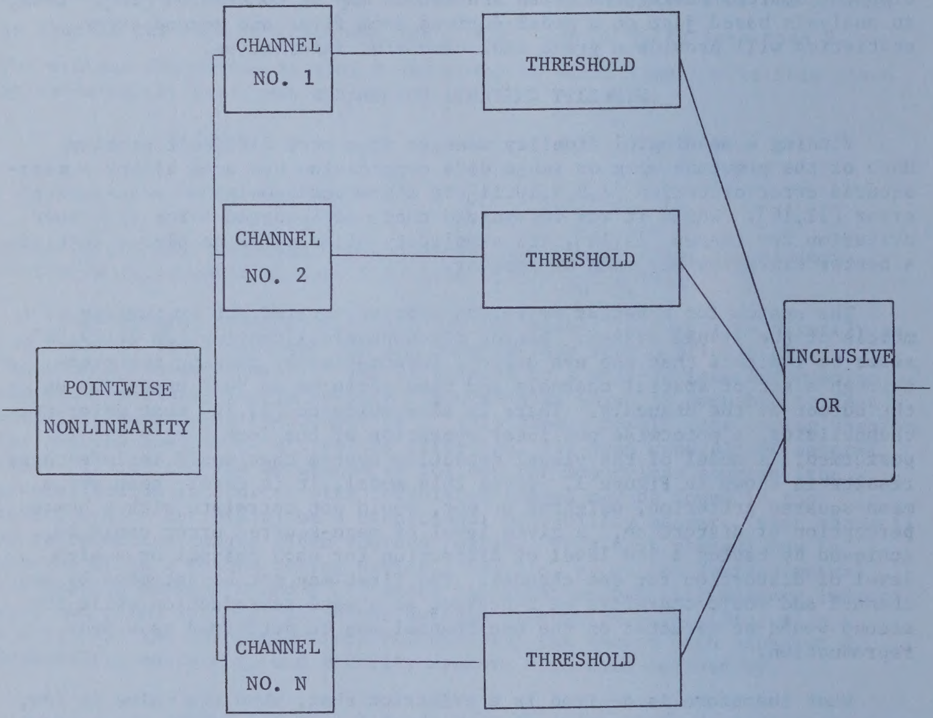

POINTWISE NONLINEARITY
CHANNEL NO. 1
CHANNEL NO. 2
CHANNEL NO. N
THRESHOLD
THRESHOLD
THRESHOLD
INCLUSIVE OR

Figure 3. Visual Detection System Model

BOUNDS TO THE DISTORTION RATE FUNCTION

While the distortion-rate function corresponding to the class of fidelity measures discussed above is well-defined and can in theory be calculated, the size of the possible input and output alphabets make this a formidable if not impossible task. It is therefore desirable to develop bounds to this distortion-rate function that are more easily calculated. These bounds may then be used to bound the performance of any actual data compression technique. Since both the linear and non-linear transformations are one-to-one, there is no distortion introduced in the transformations. We may therefore concern ourselves with the transmission of the transformed image. By determining the statistics of the transformed image, the problem may be restated as simply the determination of the distortion-rate function for a vector source with a distortion measure given by the supremum of the distortion for the individual components.

The bounds developed here are in the form of inequalities relating the distortion-rate function for a vector source with a supremum-type fidelity measure to that of scalar sources with and without side information. Since the latter distortion-rate functions may be calculated using numerical techniques [4], the resulting bounds may also be calculated and are therefore usable bounds.

For simplicity here, consider the distortion-rate function for a vector source of dimension 2. This would correspond to having two bandpass filters. The extension to arbitrary finite dimension is direct, but would result in unnecessary complexity here. Let (X, Y) be a vector source having pmf q(x,y) and marginal pmf's q(x) and q(y), where the per-letter alphabets are A_X, A_Y, and A_{XY} respectively. Let the available reproduction alphabets be $A_{\hat{X}}$, $A_{\hat{Y}}$, and $A_{\hat{X}\hat{Y}}$. Let the distortion between x and $\hat{x}$ be given by $d(x, \hat{x})$ and the distortion between y and y be $d(y, \hat{y})$. The distortion between a vector source letter (x,y) and a vector reproduction letter $(\hat{x}, \hat{y})$ is then given by

$$d((x, y), (\hat{x}, \hat{y})) = \sup (ad(x, \hat{x}), bd(y, \hat{y}))$$

where a and b are some arbitrary positive weighting constants. Define the distortion-rate functions $\delta_X(R)$, $\delta_Y(R)$, $\delta_{XY}(R)$, and $\delta_{X|Y}(R)$ to be the usual marginal and conditional distortion-rate function [1,5,17], e.g.,

$$\delta_X(R) = \inf_{p \in P} E\, d(X, \hat{X})$$

$$P = \{p(\hat{x}|x) : E \log (p(\hat{X}|X)/w(\hat{X})) \le R\}$$

$$w(\hat{x}) = \sum_x q(x)\, p(\hat{x}|x)$$

and expectation is taken with respect to the joint pmf $q(x)\, p(\hat{x}|x)$. Note that the standard coding theorem holds for all four distortion-rate functions [1,5,15,17].

Although the distortion-rate function $\delta_{XY}(R)$ may be calculated directly using numerical techniques [4], the size of the joint alphabet

A_{XY} often makes this calculation prohibitive. It is, therefore, useful to consider bounding the distortion-rate function by functions which are more readily calculable. Prime candidates for these functions are the marginal distortion-rate functions $\delta_X(R)$ and $\delta_Y(R)$ and the conditional distortion-rate function $\delta_{X|Y}(R)$. By using techniques similar to those used in deriving the relations between the joint, marginal, and conditional rate-distortion functions [15,18 sec. 3], the following upper and lower bounds to the joint distortion-rate function for the class of distortion measure considered here may be shown to hold.

Theorem: Let $\delta_X(R)$, $\delta_Y(R)$, $\delta_{X|Y}(R)$, and $\delta_{XY}(R)$ be the distortion-rate functions as defined above. Then

$$\delta_{XY}(R) \geq \inf_{\alpha \in [0,1]} \{\sup (a\delta_{X|Y}(\alpha R),\ b\delta_Y((1-\alpha)R))\}$$

$$\delta_{XY}(R_1 + R_2) \leq a\delta_X(R_1) + b\delta_Y(R_2) \ .$$

Furthermore, if we define the distortion-rate function $\delta_X^{(s)}(R)$ to be the distortion-rate function for the source and reproduction as above using a distortion measure $d_s(x,\hat{x}) = (d(x,\hat{x}))^s$, we have the upper bound for all $s \geq 1$

$$\delta_{XY}(R_1 + R_2) \leq (a^s\delta_X^{(s)}(R_1) + b^s\delta_Y^{(s)}(R_2))^{1/s} \ .$$

CONCLUSION

In this paper, the problem of compressing the data representation of an image was considered. A fidelity measure was suggested that has the tightness of a vector measure and also leads to a well-defined distortion-rate function. Bounds to this distortion-rate function that are more easily computable were developed. It is hoped that psychophysical testing of the proposed distortion measure will verify that it is indeed a criterion that reflects human perception and thus may be used in the evaluation of image compression systems.

ACKNOWLEDGEMENTS

The author would like to gratefully thank Dr. Robert M. Gray of Stanford University for his suggestions leading to this research and Roderick E. Thomas for his assistance with its computerized aspects.

REFERENCES

1. R. G. Gallager, Information Theory and Reliable Communication, New York: Wiley, 1968.

2. R. M. Gray and L. D. Davisson, "A mathematical theory of data compression?" to be published.

3. C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in 1959 IRE Nat. Conv. Rec., pt. 4, pp. 142-163.

4. R. Blahut, "Computation of channel capacity and rate-distortion functions," IEEE Trans. Inf. Theory, vol. IT-18, pp. 460-473, 1972.

5. T. Berger, Rate Distortion Theory, Englewood Cliffs, N.J.: Prentice Hall, 1971.

6. B. M. Leiner and R. M. Gray, "Rate-distortion theory for ergodic sources with side information," IEEE Trans. Inf. Theory, vol. IT-20, pp. 672-675, Sept. 1974.

7. P. A. Wintz, "Transform picture coding," Proc. IEEE, vol. 60, pp. 809-820, July 1972.

8. M. M. Sondhi, "Image restoration: the removal of spatially invariant degradations," Proc. IEEE, vol. 60, pp. 854-861, July 1972.

9. A. Habibi, "Two-dimensional Bayesian estimate of images," Proc. IEEE, vol. 60, pp. 878-883, July 1972.

10. W. K. Pratt, "Transform image coding spectrum extrapolation," Proc. Seventh Hawaii Int. Conf. on System Sciences, pp. 7-9, 1974.

11. M. Ishii et. al., "Picture bandwidth compression by DPCM in the Hadamard transform domain," Proc. Seventh Hawaii Int. Conf. on Systems Sciences, pp. 10-12, 1974.

12. L. D. Davisson, "Rate-distortion theory and application," Proc. IEEE, vol. 60, pp. 800-808, July 1972.

13. Z. L. Budrikis, "Visual fidelity criterion and modeling," Proc. IEEE, vol. 60, pp. 771-779, July 1972.

14. T. G. Stockham, Jr., "Image processing in the context of a visual model," Proc. IEEE, vol. 60, pp. 828-842, July 1972.

15. R. M. Gray, "A new class of lower bounds to information rates of stationary sources via conditional rate-distortion functions," IEEE Trans. Inf. Theory, vol. IT-19, pp. 480-489, July 1973.

16. J. L. Mannos and D. J. Sakrison, "The effects of a visual fidelity criterion on the encoding of images," IEEE Trans. Inf. Theory, vol. IT-20, pp. 525-536, July 1974.

17. B. M. Leiner, "Joint, conditional and marginal distortion-rate functions," Proc. Seventh Hawaii Int. Conf. on System Sciences, pp. 40-42, 1974.

18. B. M. Leiner, "Rate-distortion theory for sources with side information," Ph.D. dissertation, Stanford University, Stanford, California, August 1973.

19. D. J. Connor et. al., "Infraframe coding for picture transmission," Proc. IEEE, vol. 60, pp. 828-842, July 1972.

20. M. B. Sachs et. al., "Spatial-frequency channels in human vision," Jour. Optical Society of America, vol. 61, pp. 1176-1186, Sept. 1971.

21. C. F. Stromeyer, III and B. Julesz, "Spatial-frequency masking in vision: critical bands and spread of masking," Journ. Optical Society of America, vol. 62, pp. 1221-1232, Oct. 1972.

22. L. Harmon, "Masking in visual recognition: effects of two-dimensional filtered noise," Science, vol. 180, pp. 1194-1197, June 1973.

23. D. J. Sakrison, "The rate of a class of random processors," IEEE Trans. Inf. Theory, vol. IT-16, pp. 10-16, Jan. 1970.

24. R. M. Gray et. al., "A generalization of Ornstein's $\bar{d}$-distance with applications to information theory," Annals of Probability, vol. 3, no. 3, April 1975.

25. H. Mostafavi and D. J. Sakrison, "Structure and properties of a single channel in the human visual system," to be published.

ADAPTIVE DATA-COMPRESSION FOR COMMUNICATION NETWORKS*

HOWARD NUSSBAUM AND IZHAK RUBIN
Department of System Science
School of Engineering and Applied Science
University of California
Los Angeles, California 90024

ABSTRACT

Adaptive data-compression under a fidelity criterion for message sources using a communication network is considered. For such sources, generating random arriving messages at the input of a communication network, the rate-distortion function and a source coding theorem are presented. An adaptive data-compression scheme (incorporating appropriately the message queueing delays) is introduced and studied under a store-and-forward channel situation. The scheme's performance is evaluated through its Delay-Distortion function, and is found to yield considerable improvement in performance over the nonadaptive corresponding data-compression procedure.

1. Introduction

Shannon's results for the coding of sources under a fidelity criterion are normally applied to sources which continuously emit uninterrupted signal functions. For example, a discrete time source assumed to produce each unit of time a letter realization of the underlying stochastic process. When considering an information-theoretic approach to the coding of sources considered in communication network applications (as for computer and satellite communication networks), the above-mentioned source models are inadequate. An accurate representation of such a source is obtained by considering it to consist of the superposition of messages emitted at random instants of time. This paper evaluates the rate-distortion function, presents the related source-coding theorem, and considers an adaptive-compression scheme for such a source. This adaptive data-compression scheme uses information concerning the congestion of the channel to determine the amount a message is to be compressed. The delay-distortion function which was introduced in [1] is used to evaluate the scheme.

Comparing the rate-distortion function and delay-distortion function notions, we note that the rate-distortion function represents the performance of the ideal coding scheme, utilizing code-rate and reproduction-distortion as objective functions and not considering the corresponding message delays; while the delay-distortion function represents the performance of coding schemes for a specific network realization, utilizing reproduction-distortion and message delays as objective functions. We note that the message-delay function incorporated above accounts for costs due to delays accrued by the users of the network as well as for costs related to buffer storage capacities.

In a recent paper [2] the element of delay was added to the distortion function by introducing a fidelity criterion which depends upon both the coding time-delay and the accuracy of the signal reproduction. In our analysis, message-delay is used as a separate performance measure, which is appropriate for the analysis and synthesis of communication networks, where the message delay function plays a key role as a performance criteria.

---

*This work was supported by the Office of Naval Research under Contract Number N00014-75-C-0609 and by the National Science Foundation under Grant Number ENG 75-03244.

The source and network models are presented in Section II. For the information sources under consideration, emitting messages at random instants of time, the evaluation of the rate-distortion function is studied in Section III. A source coding theorem is stated, under an appropriate requirement of a finite queueing delay at the channel buffer. The adaptive data-compression scheme is presented in Section IV, and its performance analysis in terms of the Delay-Distortion function (including the relevant protocol information) is outlined and compared with that of the non-adaptive scheme.

II. Source and Network Model

The source consists of messages which arrive at random instants of time. The arrival process forms a Poisson process where the number of arrivals, $N(T)$, in a given time interval T is given by the Poisson distribution,

$$\Pr(N(T) = n) = e^{-\lambda T} \frac{(\lambda T)^n}{n!}, \quad n = 0,1,2,3\ldots,$$

where λ is the mean number of arrivals per unit time.

Messages arriving at different instants of time are assumed to be independent identically distributed and are drawn from an abstract space $\mathscr{X}$. We consider each message to be a letter $x \epsilon \mathscr{X}$, and the reproduction of x to be a letter $y \epsilon \mathscr{Y}$. The distortion accurred when letter x is reproduced by letter y is given by the distortion measure $\rho(x,y)$.

For mathematical simplicity, the network we initially consider here, for the analysis of the adaptive scheme consists of a single channel. At the input to the channel is a buffer which stores the messages prior to transmission and operates in a store-and-forward manner (thus, forwarding a message for transmission when the channel becomes free). The channel is noiseless and transmits information at the rate C(bits/sec), which we will term the channel capacity. This network model represents, for example, a satellite link, and the results and analytic approaches are characteristic of other network models as well. An application to a more complex network is given in Section IV. The discussion in Section III, regarding the Rate-Distortion function, applies to a general network model.

III. The Rate-Distortion Function for Sources Emitting Random-Occuring Messages

Considering sources emitting random-occuring messages, the two major additional elements introduced into the source-coding problem are related to the statistical characteristics of the stochastic point process of message-arrivals at the network, and of the message queueing times in the buffer facility now available to regulate the forwarding times of messages into the channels. The appropriate definition of a rate-distortion relationship, and its relation to a rate-distortion function, under the above-mentioned conditions, are presented in this section.

The computation of the rate-distortion function requires us to represent the sample function of the source over the interval $[0,T]$ by

$$\tilde{x}_T = [\{x_i\},\{t_i\},N(T)]$$

where $\{x_i\}$ is the sequence of messages (i.e., $\{x_i\} = \{x_i, i=1,2,\ldots,N(T): x_i \epsilon \mathscr{X}\}$), $\{t_i\}$ is the sequence of arrival times, and $N(T)$ is the number of messages in the interval $[0,T]$. Similarly, the sample function of the reproduction of the source is represented by

$$\tilde{y}_T \;\; = [\{y_i\},\{\hat{t}_i\},\hat{N}(T)]$$

where $\{y_i\}$ is the sequence of reproduced messages (i.e., $\{y_i\} = \{y_i, i=1,2,\ldots,\hat{N}(T): y_i \in \mathcal{Y}\}$), $\{\hat{t}_i\}$ is the sequence of reproduced arrival times, and $\hat{N}(T)$ is the number of messages reproduced. Subsequently, the rate-distortion function, $R(D)$, (see, for example, [3]) is given by

$$R(D) = \lim_{T\to\infty} \inf_{\tilde{q}_T \in \tilde{Q}_T} I(\tilde{x}_T;\tilde{y}_T) \tag{1}$$

where

$$\tilde{Q}_T = \{\tilde{q}_T : E(\rho_T(\tilde{x}_T,\tilde{y}_T)) \le D\} \tag{2}$$

and $\tilde{q}_T$ is the conditional distribution generating the underlying joint measure. Also, $\rho_T(\tilde{x}_T,\tilde{y}_T)$ is a (honest) distortion-measure, yielding the distortion resulting when the sample function $\tilde{x}_T$ is reproduced by $\tilde{y}_T$.

The average mutual information function $I(\tilde{x}_T;\tilde{y}_T)$, is thus given by

$$\begin{aligned} I(\tilde{x}_T;\tilde{y}_T) &= I([\{x_i\},\{t_i\},N(T)];[\{y_i\},\{\hat{t}_i\},\hat{N}(T)]) \\ &= I(N(T);\hat{N}(T)) + I(N(T);[\{y_i\},\{\hat{t}_i\}]|\hat{N}(T)) \\ &\quad + I([\{x_i\},\{t_i\}];N(T)|N(T)) + I(\{x_i\};\{y_i\}|N(T),\hat{N}(T)) \\ &\quad + I(\{x_i\};\{\tilde{t}_i\}|N(T),\hat{N}(T),\{y_i\}) + I(\{t_i\};\{y_i\}|N(T),\hat{N}(T),\{x_i\}) \\ &\quad + I(\{t_i\};\{\hat{t}_i\}|N(T),\hat{N}(T),\{x_i\},\{y_i\}). \end{aligned} \tag{3}$$

As in [4], we can set $\hat{N}(T) = N(T)$ since

$$\lim_{T\to\infty} \frac{1}{T} I(N(T);N(T)) = 0 .$$

Consequently, given $\hat{N}(T) = N(T)$, we have

$$\begin{aligned} R(D) = \lim_{T\to\infty} \frac{1}{T} \inf_{\tilde{q}_T \in \tilde{Q}_T} \Big\{ & I(\{x_i\};\{y_i\}|N(T),\hat{N}(T) = N(T)) \\ & + I(\{x_i\};\{\hat{t}_i\}|N(T),N(T) = \hat{N}(T),\{y_i\}) \\ & + I(\{t_i\};\{y_i\}|N(T),\hat{N}(T) = N(T),\{x_i\}) \\ & + I(\{t_i\};\{\hat{t}_i\}\; N(T),\hat{N}(T) = N(T),\{x_i\},\{y_i\}) \Big\}. \end{aligned} \tag{4}$$

The distortion measure is now taken to be of the form

$$\rho_T(\tilde{x}_T;\tilde{y}_T) = \frac{1}{T} \sum_{i=1}^{N(T)} \rho(x_i,y_i) \; , \tag{5}$$

and is thus related as indicated to the per-message distortion measure $\rho(\cdot,\cdot)$. Subsequently, the infimum in (4) occurs when $\{t_i\}$ and $\{\hat{t}_i\}$ are statistically independent and independent of $\{x_i\}$ and $\{y_i\}$. Therefore,

$$R(D) = \lim_{T\to\infty} \frac{1}{T} \inf_{q_T \in Q_T} I(\{x_i\};\{y_i\}|N(T),\hat{N}(T) = N(T)) \; , \tag{6}$$

where

$$Q_T = \{q_T : \frac{1}{T} E\left[\sum_{i=1}^{N(T)} \rho(x_i,y_i)\right] \le D\} \; .$$

Hence,

$$R(D) = \lim_{T\to\infty} \frac{1}{T} \inf_{\{D_n\}\in\mathscr{D}} E\left\{ \inf_{q_{N(T)}\in Q_{N(T)}(D_{N(T)})} I_{N(T)}(\{x_i\};\{y_i\}|N(T),\hat{N}(T) = N(T)) \right\} \tag{8}$$

where

$$Q_N(D_N) = \{q_N : E\left[\sum_{i=1}^{N} \rho(x_i,y_i)\right] \le D_N\} \text{ , } N = 1,2,\ldots, \tag{9}$$

and

$$\mathscr{D} = \left\{ D_N, N=1,2,\ldots : E\left[\frac{N(T)D_{N(T)}}{T}\right] \le D \right\}, \tag{10}$$

and q_N is the conditional distribution, given $N(T) = N$.

Setting,

$$r(D) = \inf_{q\in Q} I(\mathscr{X};\mathscr{Y}) \text{ , } 0 \le D \le D_{max} < \infty \text{ ,} \tag{11}$$

where

$$Q = \{q : E(\rho(x,y)) \le D\} \text{ ,} \tag{12}$$

we obtain

$$R(D) = \lim_{T\to\infty} \frac{1}{T} \inf_{\{D_N\}\in\mathscr{D}} E\{N(T)r(D_{N(T)})\} \text{ .} \tag{13}$$

By the convexity of $r(D)$, and by Jensen's inequality we have

$$E\left\{\frac{N(T)}{T} r(D_{N(T)})\right\} \ge \lambda r(\frac{D}{\lambda})$$

for all $\{D_N\}\in\mathscr{D}$, and all $T>0$. Therefore we conclude that

$$R(D) = \lambda r(\frac{D}{\lambda}) \text{ .} \tag{14}$$

Under a per-sample (message) distortion measure

$$\rho_T(\tilde{x}_T;\tilde{y}_T) = \frac{1}{N(T)} \sum_{i=1}^{N(T)} \rho(x_i,y_i) \text{ ,} \tag{15}$$

the corresponding rate-distortion function, $R_a(D)$, is given by

$$R_a(D) = \lim_{T\to\infty} \frac{1}{T} \inf_{\{D_N\}\in\mathscr{D}_a} E\{N(T)r(D_{N(T)})\} \text{ ,} \tag{16}$$

where

$$\mathscr{D}_a = \left\{ D_N, N=1,2,\ldots : E(D_{N(T)}) \le D \right\}. \tag{17}$$

When $r(D)$ is a continuous function, we obtain

$$R_a(D) = \lambda r(D) \tag{18}$$

In evaluating the rate-distortion function according to (6), we take a limit with respect to the time period T . Similarly, we could have taken a limit with respect to the number of messages and define a rate distortion relationship $\hat{R}(D)$ as

$$\hat{R}(D) = \lim_{N\to\infty} \inf_{\tilde{q}_N\in\tilde{Q}_N} E\left\{\frac{1}{t_N} I(\{x_i\};\{y_i\}|N)\right\}, \tag{19}$$

where

$$\tilde{Q}_N = \{\tilde{q}_N : E[\rho_{t_N}(\{x_i\};\{y_i\})] \le D\} . \tag{20}$$

In a manner analogous to that used before, we obtain

$$\hat{R}(D) = \lim_{N\to\infty} \inf_{D_t \in \hat{\mathscr{D}}} E\left[\frac{N}{t_N}\, r(D_{t_N})\right] \tag{21}$$

where

$$\hat{\mathscr{D}} = \left\{ D_t : E\left\{ \frac{D_{t_N}}{t_N} \right\} \le D \right\}. \tag{22}$$

Invoking Jensen's inequality and using the convergence results stating that $\lim_{N\to\infty} (t_N/N) = \lambda^{-1}$, a.s. , and $\lim_{N\to\infty} E(t_N/N) = \lambda^{-1}$, we obtain that

$$\hat{R}(D) = R(D) . \tag{23}$$

Defining $\hat{R}_a(D)$ in a corresponding manner, we obtain

$$\hat{R}_a(D) = R_a(D) . \tag{24}$$

Thus, the limiting procedure used in the definition of the rate-distortion function can be applied in terms of time $(\lim_{T\to\infty})$ or state $(\lim_{N\to\infty})$ block variations, under suitable conditions. For the case where $\{t_i\}$ is a general renewal point process it is many times easier to compute $\hat{R}(D)$ rather than $R(D)$.

In order to establish a source coding theorem we need to take into consideration the buffer at the input to the channel, and the resulting queueing fluctuations. We assume that if a message is present in the buffer information is transmitted out of the buffer into the network at a uniform fixed rate r .

The source coding theorem thus specifies the minimum rate of transmission out of the buffer which guarantees a finite average message delay, and shows it to be equal to $R(D)$. For a given distortion value, this interpretation of $R(D)$ is consistent with the classical situation if we allow a queueing mechanism to be placed in front of the channel. So, if the rate of transmissions, r , out of the buffer is $r > R(D)$, we can find a code of rate r and average distortion D , yielding a finite average message delay. Conversely, if $R(D) > r$, then for all codes with average distortion D , the average message delay is infinite. The source coding theorem thus states the following:

Theorem 1.

Given any $\epsilon>0$ and any $D\ge 0$, there exists a $(D+\epsilon)$-admissable block code with rate of transmission $r \le R(D)+\epsilon$ which guarantees a finite average delay for sources under consideration, where $R(D)$ is given by (14) or (18).

Outline of Proof.

Consider a coding scheme which waits for n messages to arrive and then codes these n messages. We can find a value of n large enough, such that there exists a $(D+\epsilon)$-admissable code of block length n , which encodes the n messages into m bits, where

$$m < n(r(\tfrac{D}{\lambda})+\epsilon) ,$$

by the source coding theorem [3]. Thus, the input to the buffer consists of messages of m bits in duration whose interarrival times, $\tau_i, i=1,2,\ldots$, are i.i.d. random variables governed by a gamma (Erlang) density

$$f_{\tau_i}(\tau) = \frac{(\lambda\tau)^{n-1}\lambda}{(n-1)!} e^{-\lambda\tau} , \quad \tau \geq 0 .$$

Out of the buffer the messages are transmitted at rate r. Hence, the buffer behavior follows that of an $E_n/D/1$ queueing system with the arrival process described above (see [6]), and service times equal to $\frac{m}{r}[\frac{\text{sec}}{\text{mess}}]$. From queueing theory we know that the average waiting time per block under steady state conditions is finite iff the traffic intensity ρ is less than unity, where

$$\begin{aligned} \rho &= \frac{\text{mean transmission time of a message}}{\text{mean time between arrivals}} \\ &= \frac{m}{r} / \frac{n}{\lambda} . \end{aligned}$$

If we let $r = R(D)+\epsilon = \lambda[r(\frac{D}{\lambda})+\epsilon]$, then $\rho<1$. Q.E.D.

From the proof of the theorem we observe that the buffer smooths the flow of information such that the average delay of a message becomes arbitrarily large as the excess (over D) distortion value ϵ approaches zero. Furthermore, the percentage of time that the buffer is empty, and no information is being transmitted out of it, is arbitrarily small. This implies that information is being transmitted at a uniform rate $r<R(D)+\epsilon$ almost all the time, and idle periods in transmission occur infrequently.

The converse to the coding theorem is stated as follows:

Theorem 2.

No D-admissible source code and rate of transmission r, $r<R(D)$, can yield a finite average message delay. $\blacksquare$

IV. Adaptive Data Compression: The Delay-Distortion Function

Data compression schemes for sources and channels under consideration can be evaluated and compared through their Delay-Distortion function. For a given message arrival rate λ, this function is defined as

$$\lambda_1(D) \overset{\Delta}{=} \inf_{f\in F(D)} T(\lambda;f) , \tag{25}$$

where $T(\lambda,f)$ is the delay associated with a given message arrival rate, λ, and a given bit arrival rate at the network, f. $F(D)$ is the set of all bit arrival rates at the network associated with a data compressor which yields distortion not higher than D. Considering a data compression scheme as in [1] which takes each message and compresses it into a fixed number of bits, the delay-distortion function is related to the rate-distortion relationship, $\tilde{r}(D)$, for that data compressor by

$$\lambda_1(D) = T(\lambda;\tilde{r}(D)) . \tag{26}$$

The function $T(\lambda;\tilde{r}(D))$ is expressed as the sum of two components. The first one being the average waiting time $\overline{W}$, and the second one is the transmission delay $\overline{\tau}$. The average waiting time is computed by noting that messages are data-compressed and subsequently stored in a buffer at the input of the communication channel. The latter has capacity C. Therefore, the message waiting time is that of a customer in an M/D/1 queueing system with Poisson arrivals of rate λ and message service time equal to $\frac{\tilde{r}(D)}{C}$. Using the result for average steady-state message waiting time in an M/D/1 queueing system, the average waiting is obtained as (see [5])

$$\overline{W} = \begin{cases} \frac{\tilde{r}(D)}{2C} \frac{\lambda\tilde{r}(D)}{C-\lambda\tilde{r}(D)}, & \text{if } \lambda\tilde{r}(D) < C \\ \infty, & \text{if } \lambda\tilde{r}(D) \geq C . \end{cases} \tag{27}$$

The transmission delay is given by

$$\overline{\tau} = \frac{\tilde{r}(D)}{C} . \tag{28}$$

Thus, the delay-distortion function, $\lambda_1(D)$, using a single compressor is equal to

$$\gamma_{1(D)} = \begin{cases} \frac{\tilde{r}(D)}{C} [1+\frac{1}{2} \frac{\lambda\tilde{r}(D)}{C-\tilde{r}(D)}], & \text{if } D < D_{min}(\lambda) \\ \infty, & \text{if } D \geq D_{min}(\lambda), \end{cases} \tag{29}$$

where

$$D_{min}(\lambda) = \tilde{r}^{-1}(\frac{C}{\lambda}) . \tag{30}$$

To introduce an adaptive data-compression scheme, we use instead of a single data compressor, m data compressors, each yielding a different average message distortion value. We make a decision prior to the transmission of a message as to which data compressor would operate on a given message. The decision rule is adaptive in that a decision at time t is based on the network congestion at t, and the time of arrivals of messages arriving prior to t.

A delay-distortion relationship, $\gamma_m(D)$, for a system using m compressors is now defined as

$$\gamma_m(D) = \inf_{\underline{D}\in\mathscr{D}(D)} \left\{\overline{W}_m (r(D_i),p_i,i=1,2,\ldots,m) + \sum_{i=1}^{m} \frac{p_i r(D_i)}{C} + (\text{protocal information})\right\}, \tag{31}$$

where D_i is the average distortion associated with the i^{th} data compressor, p_i is the probability of using the i^{th} data compressor, and $\overline{W}_m(\cdot)$ is the average message waiting-time. Furthermore, the set $\mathscr{D}(D)$ consists of the D_i values and adaptive decision rules (for assigning a data-compressor) which induce $\{p_i\}$ satisfying $\sum_{i=1}^{m} p_i D_i \leq D$. The protocal information term is required for specifying the data compressor used for each message.

The delay-distortion relationship using the above adaptive procedure, $\gamma_A(D)$, is given by minimizing $\gamma_m(D)$ over m, so that

$$\gamma_A(D) = \min_{m=1,2,\ldots} \gamma_m(D) .$$

The value of m that minimized $\gamma_m(D)$ is generally finite, since for a large m the channel spends a significant portion of time transmitting protocal information, and the gain achieved by the additional data compressors is lost.

The following theorem specifies the structure of the decison rule which is optimum over all decision rules under consideration, where optimality means yielding the smallest average delay, as indicated in Eq. (31). We assume that the distortion values associated with the m data compressors are ordered, $D_i < D_{i+1}, i=1,2,\ldots,m$.

Theorem 3.

The structure of the stationary decision rule which gives the smallest average delay for m data compressors, characterized by $(D_i, p_i) i=1,2,\dots,m$, is such that it chooses data compressor i for the j^{th} message if and only if the waiting time, W_j, for the j^{th} message satisfies

$$T_{i-1} < W_j < T_i ,$$

where $T_i, i=1,2,\dots,m-1$, are thresholds chosen to minimize the average delay.∎

Thus, the optimum decision rule uses m-1 thresholds, $\{T_i, i=1,\dots,m-1\}$ to compare the actual waiting times with the thresholds, as depicited in Fig. 1.

In order to incorporate protocal information in the model, we assume that each message requires $\log_2(m)$ bits to specify the data compressor used. Thus, a message assigned to a data compressor yielding distortion value D, will contain $\tilde{r}(D) + \log_2(m)$ bits when it is transmitted over the channel.

Defining $\gamma_m(\{D_i\},\{T_i\})$ as the average message delay when we have m data compressors with distortions $D_i, i=1,2,\dots,m$ and thresholds $T_i, i=1,2,\dots,m-1$, we find that

$$\gamma_m(D) = \inf_{\{D_i\},\{T_i\}\in\mathscr{D}_m(D)} \gamma_m(\{D_i\},\{T_i\}) \tag{32}$$

where

$$\mathscr{D}_m(D) = \{\{D_i\},\{T_i\};\ \sum_{j=1}^{m} p_j(\{D_i\},\{T_i\}) D_i \le D\}, \tag{33}$$

and $p_j(\{D_i\},\{T_i\})$ is the probability of using compressor j given distortion values $\{D_j\}$ and thresholds $\{T_i\}$. To compute the functions $\gamma_m(\{D_i\},\{T_i\})$ and $p_j(\{D_i\},\{T_i\})$ for the scheme under consideration, we use techniques from queueing theory. We first consider the imbedded markov chain which consists of the number of messages compressed by the various data compressors which are in the buffer when a message leaves the channel. The state of the buffer at these instants of time is described by an m-dimensional vector $\underline{i}$, whose components are the number of messages from the various data compressors.

We let $p_{\underline{j}|\underline{i}}$ be the stationary transition matrix for the markov chain, and $\pi_{\underline{i}}$ be the steady-steady probability distribution obtained as the solution to the set of equations,

$$\pi_{\underline{j}} = \sum_{\underline{i}} p_{j|i}\, \pi_{\underline{i}} . \tag{34}$$

We denote by $W_{\underline{j}|\underline{i}}$ the sum of the average waiting times of messages arriving during the time the system transitions from state $\underline{i}$ to state $\underline{j}$, and $N^{\ell}_{\underline{j}|\underline{i}}$ represents the number of messages assigned to data compressor ℓ during the time the system transitions from state $\underline{i}$ to state $\underline{j}$. By ergodicity of the markov chain, the probability $p_\ell, \ell=1,2,\dots,m$, is given by

$$p_\ell = \frac{\sum\limits_{\underline{i}} \sum\limits_{\underline{j}} N^\ell_{\underline{j}|\underline{i}}\, p_{j|i}\, \pi_i}{\sum\limits_{k=1}^{m} \left(\sum\limits_{\underline{i}} \sum\limits_{\underline{j}} N^k_{\underline{j}|\underline{i}}\, p_{\underline{j}|\underline{i}}\, \pi_{\underline{i}}\right)}, \tag{35}$$

and the average waiting time $\overline{W}$ is given by

$$\overline{W} = \frac{\sum\limits_{\underline{i}} \sum\limits_{\underline{j}} W_{\underline{j}|\underline{i}}\, p_{\underline{j}|\underline{i}}\, \pi_{\underline{i}}}{\sum\limits_{k=1}^{m} \left(\sum\limits_{\underline{i}} \sum\limits_{\underline{j}} N^k_{\underline{j}|\underline{i}}\, p_{\underline{j}|\underline{i}}\, \pi_{\underline{i}}\right)} .$$

Subsequently, the delay function $\gamma_m(\{D_i\},\{T_i\})$ is given by

$$\gamma_m(\{D_i\},\{T_i\}) = \overline{W} + \frac{\log_2(m)}{C} + \sum_{i=1}^{m} p_i \frac{\tilde{r}(D_i)}{C} \tag{36}$$

Furthermore, we have derived analytic expressions for $p_{\underline{j}|\underline{i}}$, $\pi_{\underline{i}}$, $N_{\underline{j}|\underline{i}}$ and $W_{\underline{j}|\underline{i}}$, and incorporating them in the above expressions, we have developed a numerical technique to compute $\overline{W}$ and p_ℓ.

Example.

Consider the case where messages consist of 100 i.i.d. gaussian random variables with variance σ^2 and arrival rate λ of 1000 messages/sec. We let

$$\tilde{r}(D) = 100 \cdot R(D)$$

where $R(D)$ is the rate-distortion for a gaussian random variable. Using numerical techniques to evaluate $\gamma_m(\{D_i\},\{T_i\})$, $\gamma_m(D)$ was computed for m=1,2,3 and Fig. 2 presents these curves. From the figure, the best Delay-Distortion relationship achieved by schemes using 3 or fewer data compressors is found to be the lower envelope of the curves. For high distortions, schemes with one data compressor are optimum, while as the distortion decreases more data compressors are required. In fact using more than one data compressor we can achieve a finite delay for distortion values less than D_{min}, which is the minimum distortion value giving a finite value for $\gamma_1(D)$.

A Tandem Network

Consider a network consisting of n channels in tandem as depicted in Fig. 3, where each channel has capacity C. This network model represents the transmission mechanism for messages transmitted from a source to a destination in a complex communication network where the effects of other messages from other sources are negligible. The nodes in the model operate in a store-and-forward manner; i.e. retransmission of a message over a link only occurs after the totality of the message or submessage (packet) is assembled by the preceding node, and the required buffers for queueing messages are available in all the nodes.

In order to compute the delay-distortion relation $\gamma_m(\{D_i\},\{T_i\})$ for this network, we must first compute the average waiting time at each node. The following theorem gives the average waiting time at nodes other than the first node.

Theorem 4.

The waiting time $W_{(\ell)}$ of a message, compressed with the ℓ^{th} compressor, at a node in the network other than the first node, is given by

$$W_{(\ell)} = \frac{\tilde{r}(D_1)-\tilde{r}(D_\ell)}{C} . \blacksquare \tag{38}$$

Using Theorem 4, $\gamma_m(\{D_i\},\{T_i\})$ is obtained and is given by

$$\gamma_m(\{D_i\},\{T_i\}) = \overline{W} + n \frac{\log_2(m)}{C} + (n-1) \frac{\tilde{r}(D_1)}{C} + \sum_{j=1}^{m} p_j(\{D_i\},\{T_i\}) \frac{r(D_j)}{C} \tag{39}$$

Hence, the analysis of the adaptive data-compression scheme for this network model is essentially identical to that presented above for a single channel network, with the simple modifications in the expression for the transmission time, as shown in (39).

REFERENCES

1. I. Rubin, "Data Compression for Communication Networks: The Delay-Distortion Function", presented at the *1974 Symposium on Information Theory*, Notre Dame, Indiana and submitted for publication.

2. S. I. Krich and T. Berger, "Coding for a Delay-Dependent Fidelity Criterion", *IEEE Trans. Inform. Theory*, Vol. IT-20, pp. 77-85, Jan., 1974.

3. T. Berger, *Rate Distortion Theory-A Mathematical Basis for Data Compression*, Englewood Cliffs, N.J., Prentice Hall, 1971.

4. I. Rubin, "Information Rates and Data Compression Schemes for Poisson Processes", *IEEE Trans. Inform., Theory*, Vol. IT-20, pp. 200-210, March, 1974.

5. I. Rubin, "Communication Networks: Message Path Delays", *IEEE Trans. Inform. Theory*, Vol. IT-20, pp. 738-745, Nov., 1974.

6. J. W. Cohen, *The Single Server Queue*, North Holland Publishing Co., 1969.

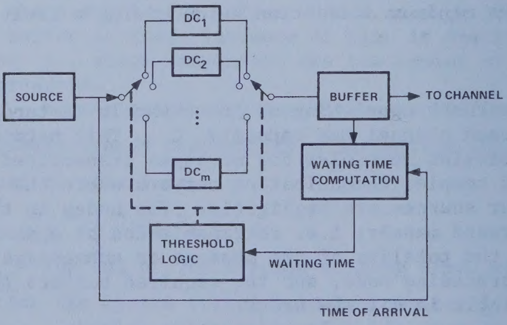


Figure 1. Data Compression Scheme.

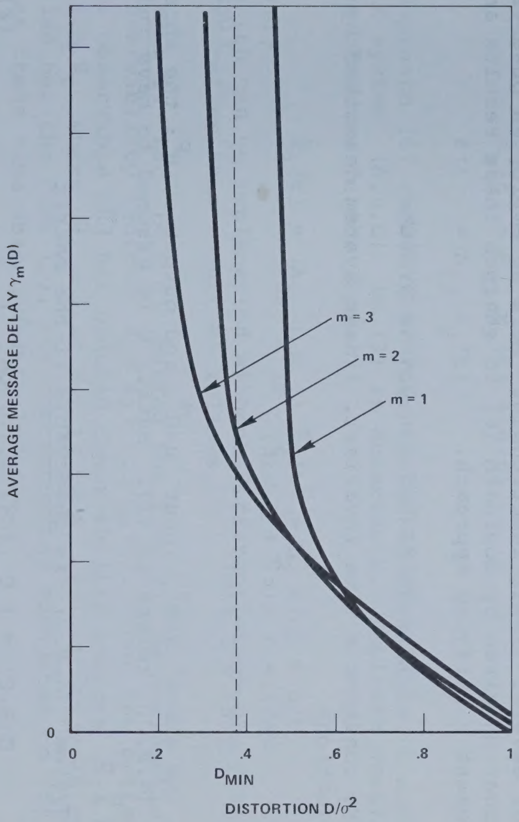


Figure 2. Delay–Distortion Curves for the Single Channel System Gaussian Source, Number of Data Compressors = m.

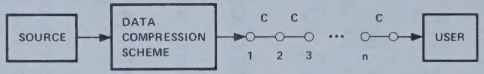

SOURCE
DATA COMPRESSION SCHEME
C
C
C
1
2
3
...
n
USER

Figure 3. Network Consisting of Tandem Channels.

OBSERVERS OF LINEAR FUNCTIONALS*

BRADLEY W. DICKINSON
Department of Electrical Engineering
Princeton University, Princeton, New Jersey 08540

ABSTRACT

The structure of observers of a set of linear functionals of the state of a linear system is studied by extending an approach due to Gopinath. This clarifies and refines the previous results known for this problem.

I. INTRODUCTION

Observers for linear systems were introduced by Luenberger [1] as a means of state estimation based on measurable inputs and outputs. In applications where a linear state variable control law is to be implemented, it is often possible to reduce the order of the observer required to produce estimates of the desired linear functionals. For the case of a single linear functional, the minimal order observer, having arbitrary poles, was described by Luenberger [1] and more rigorously by Wonham and Morse [2]. Fortmann and Williams [2], Murdoch [4], and Barry and Waters [5] have looked at design procedures for the case of several linear functionals. Here we employ an observer construction given by Gopinath [6] to sharpen these results and to present a unifying approach.

II. OBSERVERS FOR LINEAR SYSTEMS

We consider a time invariant, linear system described by the equations

$$\dot{x}(t) = A\,x(t) + B\,u(t) \qquad (1a)$$

$$y(t) = C\,x(t) \qquad (1b)$$

where the state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$, and output $y \in \mathbb{R}^p$; the shorthand (A,B,C) refers to (1). Matrix C is assumed to have full rank p.

System (A,B,C) is observable if the matrix

$$\mathcal{O}(A,C) = [\, C', (CA)', \ldots, (CA^{n-1})' \,]' \qquad (2)$$

(prime denotes transpose), has rank n. The observability indices of system (A,B,C) are defined in the following convenient way. The i^{th} index, s_i, is the least integer such

---

* This research was supported by NSF Grant No. ENG75-10533.

that row $p(s_i-1)+i$ of $\mathcal{O}(A,C)$ is linearly dependent on the rows above it in $\mathcal{O}(A,C)$. We assume that the outputs are arranged so that $s_1 \geqslant s_2 \geqslant \ldots \geqslant s_p > 0$. The ordered observability indices are invariant under transformations taking system (A,B,C) to system $(T^{-1}(A+LC)T, T^{-1}B, CT)$ [2]. Finally, we recall that the invariant factors of system (A,B,C) are the non-unit invariant polynomials of the matrix $(sI-A)$, ordered in decreasing degree. These are a convenient characterization of the dynamics of system (A,B,C) in the familiar way.

Since C has rank p, we may take the form of (1) to be

$$\begin{aligned} \dot{x}_1(t) &= A_{11}\, x_1(t) + A_{12}\, x_2(t) + B_1\, u(t) \\ \dot{x}_2(t) &= A_{21}\, x_1(t) + A_{22}\, x_2(t) + B_2\, u(t) \\ y(t) &= x_1(t) \end{aligned} \tag{3}$$

Clearly only the states $x_2(t)$ need be estimated by any observer. With

$$\begin{aligned} z(t) &= \dot{y}(t) - A_{11}\, y(t) - B_1\, u(t) \\ \bar{u}(t) &= A_{21}\, y(t) + B_2\, u(t) \end{aligned} \tag{4}$$

the equations for $x_2(t)$ can be rewritten as

$$\begin{aligned} \dot{x}_2(t) &= A_{22}\, x_2(t) + \bar{u}(t) \\ z(t) &= A_{12}\, x_2(t) \end{aligned} \tag{5}$$

Gopinath [6] showed that this system is observable if and only if system (A,B,C) in (3) is observable. Gopinath constructed an observer in the form

$$\dot{\hat{x}}_2(t) = (A_{22}-L_2A_{12})\, \hat{x}_2(t) + L_2\, z(t) + \bar{u}(t) \tag{6}$$

which can be implemented without differentiation of $y(t)$. We can refine this result as follows.

Lemma 1. Let (A,B,C) having the form (3) be observable with observability indices $s_1 \geqslant s_2 \geqslant .. \geqslant s_p > 0$. Then the system (5) is observable and has ordered observability indices $\hat{s}_1 \geqslant \hat{s}_2 \geqslant \ldots \geqslant \hat{s}_p$, where $\hat{s}_i = s_i - 1$ for $1 \leqslant i \leqslant p$. If $(T^{-1}AT, T^{-1}B, CT)$ also has the form (3), the corresponding subsystem to (5) also has these same observability indices.

Proof. Taking system (A,B,C) in Luenberger's "canonical form" [1] suitably reordered gives the first result. For T to preserve the form (3), it must satisfy

$$T = \begin{bmatrix} I_p & 0 \\ T_1 & T_2 \end{bmatrix}, \ \det T_2 \neq 0 \tag{7}$$

The induced transformation of system (5) is $(T_2^{-1}(A_{22}-T_1A_{12})T_2, T_2^{-1}, A_{12}T_2)$ and so the observability indices are preserved.

Theorem 1. Dynamical Structure of Observers. Let system (A,B,C) in (3) be observable with $s_1 \geq s_2 \geq \dots \geq s_p > 0$ its observability indices. By choice of L_2, the invariant factors of the system (6) can be chosen to be any monic polynomials with real coefficients, $\phi_1(s), \dots, \phi_q(s)$, $q \leq p$, satisfying

i) $\phi_{i+1}(s)$ divides $\phi_i(s)$, $1 \leq i \leq q-1$

ii) $$\sum_{i=1}^{k} \deg \phi_i(s) \geq \sum_{i=1}^{k} (s_i - 1), \quad 1 \leq k \leq p$$

Proof. The result follows from Lemma 1 by applying Rosenbrock's "control structure theorem" [7] in its dual form. In practice, $\phi_1(s)$, the minimal polynomial of $A_{22}-L_2A_{12}$, is chosen to have "stable" roots and then (6) functions as an observer.

III. OBSERVING LINEAR FUNCTIONALS

Let $w(t)$ be some desired linear functionals of the state of (3) expressed as

$$w(t) = M\, x_1(t) + N\, x_2(t) \tag{8}$$

where N is an $r \times (n-p)$, rank r matrix. To observe $w(t)$, it suffices to observe $N\, x_2(t)$. Let

$$\hat{w}_2(t) = N\, \hat{x}_2(t) \tag{9}$$

where $\hat{x}_2(t)$ satisfies (6). As a system (6,9) will not be observable if q, the number of its invariant factors, is greater than r. In many practical cases, we often have $r \leq m < p$, so that if L_2 in (6) was chosen to give p invariant factors, considerable reduction in the order of the observer can be made. This can be done simply by discarding the unobservable states. More precisely, if $\phi_1(s), \dots, \phi_r(s)$ are the first r invariant factors of (6,9) and if $\psi_1(s), \dots, \psi_q(s)$ are the invariant factors of the reduced system (here $q \leq r$), then $\psi_i(s)$ divides $\phi_i(s)$ for $1 \leq i \leq q$. In the worst case, which is also the generic or most likely case, $\psi_i(s) = \phi_i(s)$ for $1 \leq i \leq q$. Thus the bounds of Theorem 1 apply.

Theorem 1 essentially guarantees that an observer having as its minimal polynomial any (s_1-1)-th degree monic polynomial with real coefficients can be constructed. Actually we may need to augment (6) with additional observable but not controllable states to increase odd observability indices in order to obtain invariant factors with real coefficients; also it may be necessary to augment the observer obtained after unobservable states are discarded. This latter step would never be done in practice.

While the highly non-cyclic form of the proposed observer may be desirable for parallel implementation or for getting finite settling-time observers for discrete-time systems, we would generally like to know if such constraints are essential.

It will prove convenient to use the flexibility inherent in Lemma 1 to provide a simple form for A_{22} and A_{12} in (6). We may use the dual "Brunovsky form", see [2], where

$$A_{22} = \left\{ \text{block diag.} \begin{bmatrix} 0 & \dots & 0 \\ 1 & & \vdots \\ \cdot & & \vdots \\ 0 & \dots 1 & 0 \end{bmatrix} ; \ s_i-1 \times s_i-1 \right\} \quad (10a)$$

$$A_{12} = \left\{ \text{block diag.} [\ 0 \dots 0 \ 1 \ ]; \quad 1 \times s_i-1 \right\} \quad (10b)$$

In (10) we have implicitly neglected the unit observability indices, since they will play no role. Notice that with A_{22} and A_{12} in this form, it is particularly easy to choose L_2 in (6) so that the degrees of the invariant factors of (6,9) are equal to the $\{ s_i-1 \}$. The form (10) also induces a conformable partition of the state vector and the matrix N in (9) in the obvious way. Notice that any of these substate vectors having a zero multiplier in the partition of N can be disregarded in what follows without loss of generality.

Theorem 2. Let N be a $1 \times (n-p)$ matrix and let $\chi(s)$ be a monic polynomial of degree s_1-1 with real coefficients. There is an observer of the linear functional associated with N having $\chi(s)$ as its characteristic polynomial.

Proof. Observable but uncontrollable states are easily added to (6,9) when it has the form (10) so that all observability indices are made equal to s_1-1. By Theorem 1, choose L_2 to give $\phi_i(s) = \chi(s)$ for $1 \leq i \leq p$. After unobservable states are deleted from (6,9), we are left with a system whose characteristic polynomial divides $\chi(s)$. Additional states can be added to obtain equality if necessary; see [2] for a precise statement to this effect.

Of course, this is Luenberger's original result obtained by this new approach. It is easily shown that given N, almost all L_2 (in the generic sense) cause the first s_1-1 columns of the observability matrix of (6,9) to be linearly independent. In this case, the state of the observer can be taken to be $\hat{x}_1$, the first substate vector in the partition of $\hat{x}$

corresponding to (10), modified by the addition of a particular linear combination of the remaining substate vectors.

Recognizing just what the state of the observer of a single linear functional is allows one to simplify observers designed for further linear functionals. This idea was mentioned by Murdoch [4], but Barry and Waters [5] presented the following kind of result. To observe a second linear functional, the state of the observer of the first one can be used to eliminate the dependence of the second on $\hat{x}_1$. Then an observer for the modified functional can be constructed independently with a characteristic polynomial of degree s_2-1. Examples show that it may be possible to further reduce the order of the overall observer of the two functionals by allowing the states of the second one to feed into the states of the first in a manner compatible with the equations (6). One might then conjecture that a minimal order observer could be constructed this way, a somewhat modified version of a conjecture of Fortmann and Williams [3]. This remains one of the interesting open problems of this research area.

The extension of this idea to more than two functionals is straightforward. Unfortunately, this is not quite enough freedom to assure that the characteristic polynomial of the overall observer can be any monic polynomial having degree s_1+s_2-2; if both are even, the observer must have two real poles. The restriction of Murdoch [4] about poles of the observer and the original system being disjoint has been removed.

A transfer function approach to the observer design problem can also be taken. This was noted by Chen [8] for the case $m=p=1$ and more generally by Wolovich [9]. If we take the transform of the differential equations of the observer, we obtain the frequency domain equation

$$w(s) = H_u(s)\ u(s) + H_y(s)\ y(s) \tag{11}$$

We will take the case of most interest for implementation of a state variable control law, namely that the observed linear functionals are to be used as an input signal so that

$$y(s) = H(s)\ w(s) \tag{12}$$

where $H(s)$ is the transfer function of the original system. Using a right matrix fraction representation of the system transfer function,

$$H(s) = N(s)\ D^{-1}(s) \tag{13}$$

and using left matrix fraction descriptions of $H_u(s)$ and $H_y(s)$

$$H_u(s) = D_o^{-1}(s)\, N_u(s) \tag{14}$$

$$H_y(s) = D_o^{-1}(s)\, N_y(s) \tag{15}$$

see [9] for the details, the solution for the observer transfer functions is seen to be given by the solution of the matrix diophantine equation

$$N_y(s)\, N(s) + (\, D_o(s) + N_u(s)\,)\, D(s) = D_o(s)\, D_c(s) \tag{16}$$

where the desired closed-loop transfer function is

$$H_c(s) = N(s)\, D_c^{-1}(s) \tag{17}$$

Applying a recent result of Rosenbrock and Hayton [11], it can be shown that for a certain class of "observer denominators", $D_o(s)$, the transfer functions $H_u(s)$ and $H_y(s)$ obtained as a solution to (16) are proper rational matrices and hence, by Wolovich's correspondence between these problems [9], their realizations are indeed observers of the required linear functionals. Although there is not yet a complete state space development for the following result, there is a transfer function approach as outlined above.

Theorem 3. Let $\phi_1(s), \ldots, \phi_q(s)$, $q \leq m$, be any monic polynomials satisfying

i) $\phi_{i+1}(s)$ divides $\phi_i(s)$ $1 \leq i \leq q$

ii) $\sum_{i=1}^{k} \deg \phi_i(s) \geq \sum_{i=1}^{k} (s_i - 1)$ $1 \leq k \leq m$

Then there is an observer of any m linear functionals whose invariant factors are given by the $\{\phi_i(s)\}$.

It is felt that a great deal more insight into the observer design problem can come from a deeper study and understanding of the connections between the state space and transfer function approaches.

REFERENCES

1. D.G. Luenberger, "Observers for Multivariable Systems", IEEE Trans. on Automat. Contr., AC-11, pp. 190-197, 1966.

2. W.M. Wonham and A.S. Morse, "Feedback Invariants of Linear Multivariable systems", Automatica, 8, pp. 93-100, 1972.

3. T.E. Fortmann and D. Williamson, "Design of Low-Order Observers for Linear Feedback Control Laws", IEEE Trans. on Automat. Contr., AC-17, pp. 301-308, 1972.

4. P. Murdoch, "Design of Degenerate Observers", IEEE Trans. on Automat. Contr., AC-19, pp. 441-442, 1974.

5. P.E. Barry and C.R. Waters, "Multifunction Observers", submitted for publication.

6. B. Gopinath, "On the Control of Linear Multiple Input-Output Systems", Bell Syst. Tech. J., 50, pp. 1063-1081, 1971.

7. H.H. Rosenbrock, State-Space and Multivariable Theory, Wiley, 1970, see p. 190.

8. C.T. Chen, "A New Look at Transfer Function Design", Proc. IEEE, 11, pp. 1580-1585, 1971.

9. W.A. Wolovich, Linear Multivariable Systems, Springer-Verlag, 1974.

10. H.H. Rosenbrock and G.E. Hayton, "The General Problem of Pole Assignment", Control Systems Centre Report No. 288, Univ. of Manchester Institute of Science and Technology, 1975.

ON THE IRREDUCIBLE JORDAN CANONICAL FORM REALIZATION

YASUYUKI FUNAHASHI
Automatic Control Laboratory,
Nagoya University
Chikusaku, Nagoya 464, Japan

ABSTRACT

A new and straightforward method for realizing a rational transfer-function matrix into an irreducible Jordan canonical form state equation is presented. The simple operation determines A and B matrices simultaneously. The C matrix is then determined as a unique solution of a linear algebraic equation. The presented algorithm is much simpler than the hitherto reported ones.

1. INTRODUCTION

The realization of a rational transfer-function matrix into an irreducible state equation is one of the fundamental problems in linear system theory and has been discussed by many authors. Among them the realization into the irreducible Jordan canonical form has been studied from the early stage. A method for realizing a rational matrix with simple poles into an irreducible Jordan canonical form state equation is well known [1] ∿ [3]. For a rational matrix with multiple poles, Kalman [4] proposed a method by using the Smith - McMillan canonical form, which is not easy to obtain in general. Later, several methods have been proposed from the computational point of view. However, the method given by Panda and Chen [5] is indirect and in general requires a number of nonsingular transformations in order to transform the system into an irreducible form. Kuo [6] gave a method which determines A matrix in a neat way but is rather complex in computing B and C matrices. The methods by Lal et al [7] and Huang [8] are rather involved.

In this paper, we propose a new and straightforward method for realizing a rational matrix into an irreducible Jordan canonical form state equation. The basic ideas rely on the facts : (i) For the Jordan canonical form state equation, the criteria of controllability and observability are simple and specific, and (ii) a Jordan matrix with eigenvalue 0 acts as a kind of shift operator. The method consists of two steps. First, we form an augmented matrix from the coefficient matrices of the given rational matrix and extract linearly independent vectors from the augmented matrix in a specified way. This operation alone determines the A and B matrices of an irreducible Jordan canonical form realization. Second, the C matrix is determined as a unique solution of the linear matrix equation. Our method requires no concepts other than the operation of choosing linearly independent vectors. It is convenient for both machine and hand computations.

We describe the realization problem in Section 2 where the reformulation of the problem is also given. In Section 3 we sketch the system theoretic features possessed by Jordan canonical form state equation. A realization algorithm is presented in Section 4 and some numerical examples are given in Section 5 to illustrate the procedure.

2. REALIZATION PROBELM IN THE JORDAN CANONICAL FORM

Given an $m \times p$ rational transfer matrix $G(s)$, a realization of

$G(s)$ is a quadruple of matrices $\{A, B, C, D\}$ satisfying the relation

$$C(sI - A)^{-1}B + D = G(s) \qquad \text{for all } s. \tag{2.1}$$

A, B, C and D are of dimensions $n \times n$, $n \times p$, $m \times n$ and $m \times p$, respectively. A realization is called irreducible if n is equal to the degree of $G(s)$ [4]. Since D is determined as $\lim_{s\to\infty} G(s)$, we shall assume that $G(s)$ is a strictly proper rational matrix. It is further assumed, without loss of generality, that $G(s)$ contains only one pole, e.g., $s = \lambda$ with multiplicity r. Let $G(s)$ be expanded into partial fractions.

$$G(s) = \frac{G_0}{s-\lambda} + \frac{G_1}{(s-\lambda)^2} + \dots + \frac{G_{r-1}}{(s-\lambda)^r} \tag{2.2}$$

Thus the realization problem is reduced to that of finding $\{A, B, C\}$ with $\dim A$ minimal, satisfying

$$C(sI - A)^{-1}B = \frac{G_1}{s-\lambda} + \frac{G_2}{(s-\lambda)^2} + \dots + \frac{G_{r-1}}{(s-\lambda)^r} \tag{2.3}$$

such that

$$A = \lambda I_n + J \tag{2.4}$$

where I_n denotes an $n \times n$ identity matrix and J is an $n \times n$ matrix of the form

$$J = \begin{bmatrix} 0 & \alpha_1 & 0 & & & \\ 0 & 0 & \alpha_2 & 0 & \dots & \\ & \cdot & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \alpha_{n-1} \\ & & & & & 0 \end{bmatrix} \tag{2.5}$$

and $\alpha_i = 0$ or 1 for $i = 1, 2, \dots, n-1$.

Since

$$J^{\ell} = 0 \qquad \ell = n, n+1, n+2, \dots \tag{2.6}$$

$C(sI - A)^{-1}B$ can be expanded as follows

$$C(sI - A)^{-1}B = \sum_{\ell=1}^{n} \frac{CJ^{\ell-1}B}{(s-\lambda)^{\ell}}. \tag{2.7}$$

The comparison of (2.3) and (2.7) enables us to reformulate the present problem.

Realization Problem   Given coefficient matrices $G_0\ G_1 \dots G_{r-1}$ of (2.3), determine $\{J, B, C\}$ with J of the form (2.5) and $\dim J$ minimal, satisfying the relation

$$\left.\begin{aligned} CB &= G_0 \\ CJB &= G_1 \\ &\vdots \\ CJ^{r-1}B &= G_{r-1} \\ CJ^{\ell}B &= 0 \qquad \ell=r, r+1, \dots \end{aligned}\right\} \tag{2.8}$$

3. JORDAN CANONICAL FORM STATE EQUATIONS

Consider the system described by $\{J, B, C\}$. We form $\bar{J}$ from J by arranging the Jordan blocks of the same size through row and column elementary operations. For example, if J has two Jordan blocks of size 3 , $\bar{J}$ is of the form

$$\bar{J} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Such a transformation is nonsingular and is represented by

$$\bar{J} = TJT^{-1} \tag{3.1}$$

where T is a nonsingular matrix. Define

$$\bar{B} = TB$$
$$\bar{C} = CT^{-1}$$

then $\{\bar{J}, \bar{B}, \bar{C}\}$ and $\{J, B, C\}$ are equivalent, i.e.,

$$CJ^jB = \overline{CJ^jB} \tag{3.2}$$

Now we consider the system $\{\bar{J}, \bar{B}, \bar{C}\}$ with k_i Jordan blocks of size $(i+1)$, $i=0, 1, 2, \ldots, r-1$.

$$\bar{J} = \left[\begin{array}{ccccc|c|ccc|cc|c}
0 & I_{k_{r-1}} & 0 & & & & & & & & & \\
0 & 0 & I_{k_{r-1}} & 0 & & & & & & & & \\
 & & 0 & \ddots & & & & & & & & \\
 & & & \ddots & I_{k_{r-1}} & & & & & & & \\
 & & & & 0 & & & & & & & \\
\hline
 & & & & & \ddots & & & & & & \\
\hline
 & & & & & & 0 & I_{k_2} & 0 & & & \\
 & & & & & & 0 & 0 & I_{k_2} & & & \\
 & & & & & & 0 & 0 & 0 & & & \\
\hline
 & & & & & & & & & 0_{k_1} & I_{k_1} & \\
 & & & & & & & & & 0_{k_1} & 0_{k} & \\
\hline
 & & & & & & & & & & & 0_{k_0}
\end{array}\right] \qquad \bar{B} = \left[\begin{array}{c} B_0^r \\ B_1^{r-1} \\ \vdots \\ B_{r-1}^1 \\ \hline \vdots \\ \hline B_0^3 \\ B_1^2 \\ B_2^1 \\ \hline B_0^2 \\ B_1^1 \\ \hline B_0^1 \end{array}\right]$$

$$\bar{C} = \left[\begin{array}{ccccc|c|ccc|cc|c} C_{r-1}^1 & C_{r-2}^2 & \cdots & C_1^{r-1} & C_0^r & \cdots & C_2^1 & C_1^2 & C_0^3 & C_1^1 & C_0^2 & C_0^1 \end{array}\right]$$

The observability and controllability of $\{J, B, C\}$ is easily judged by $\bar{C}$ and $\bar{B}$, respectively.

Lemma 1 (i) $\{J, B\}$ is a controllable pair if and only if rows of $\begin{bmatrix} B_{r-1}^1 \\ \vdots \\ B_1^1 \\ B_0^1 \end{bmatrix}$ are linearly independent.

(ii) $\{J, C\}$ is an observable pair if and only if the columns of $[C^1_{r-1}\ C^1_{r-2} \dots C^1_2\ C^1_1\ C^1_0]$ are linearly independent.

This lemma is a restatement of [9]. As for the $\overline{J}$, we have the following interpretation.

Lemma 2 The $\overline{J}$ acts as a kind of shift operator, i.e. ,

$$\overline{J}\,\overline{B} = \begin{bmatrix} B_1^{r-1} \\ \vdots \\ B_{r-1}^1 \\ 0 \\ \cdots \\ \\ \cdots \\ B_1^2 \\ B_2^1 \\ 0 \\ \cdots \\ B_1^1 \\ 0 \\ \cdots \\ 0 \end{bmatrix}, \quad \overline{J}^2\overline{B} = \begin{bmatrix} B_2^{r-2} \\ \vdots \\ B_{r-1}^1 \\ 0 \\ 0 \\ \cdots \\ \\ \cdots \\ B_2^1 \\ 0 \\ 0 \\ \cdots \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}, \quad \overline{J}^3\overline{B} = \begin{bmatrix} B_3^{r-3} \\ \vdots \\ B_{r-1}^1 \\ 0 \\ 0 \\ 0 \\ \cdots \\ \\ \cdots \\ 0 \\ 0 \\ 0 \\ \cdots \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}, \dots \qquad (3.3)$$

We now define following augmented matrices :

$$H_j = \begin{bmatrix} G_j & G_{j+1} & \cdots & G_{r-1} & \\ G_{j+1} & G_{j+2} & \cdots & G_{r-1} & 0 \\ \vdots & & \iddots & & \\ G_{r-2} & G_{r-1} & 0 & & \\ G_{r-1} & 0 & & & \end{bmatrix} \qquad j = 0, 1, \dots, r-1 \qquad (3.4)$$

Theorem 1 If the system $\{J, B, C\}$ is both controllable and observable, the rank of H_j , denoted by n_j , is related through the number k_i of Jordan blocks of size $(i+1)$, $i=o, 1, \dots, r-1$, by

$$n_j = k_j + 2k_{j+1} + 3k_{j+2} + \dots + (r-j)k_{r-1}\ , \quad j=0, 1, .., r-1. \qquad (3.5)$$

Conversely, the k_i is given by

$$k_i = n_i - 2n_{i+1} + n_{i+2}\ , \quad i=0, 1, .., r-1 \qquad (3.6)$$

where it is understood that $n_r = n_{r+1} = 0$.

(Proof) From (2.8) (3.2) and (3.4), we have

$$\text{rank } H_j = \text{rank} \begin{bmatrix} \overline{C} \\ \overline{CJ} \\ \vdots \\ \overline{C}\ \overline{J}^{r-1} \end{bmatrix} \left[\overline{J}^j\overline{B},\ \overline{J}^{j+1}\overline{B},\ ..,\ \overline{J}^{r-1}\overline{B}\right]$$

$$= \text{rank} \left[\overline{J}^j\overline{B},\ \overline{J}^{j+1}\overline{B},\ \dots,\ \overline{J}^{r-1}\overline{B}\right] \qquad (3.7)$$

where the observability assumption is used. Then (3.5) is obvious from Lemma 1, 2 and $k_i = \text{rank } B_i^1$.

Remark : The relations (3.5) and (3.6) were derived in [6] in a different way. We note the special appearance of B_i^j's in (3.7), which plays a key role in the algorithm to be described.

4. REALIZATION ALGORITHM

In the sequel it is assumed $r=4$ for the simplicity of presentation. The general case can be treated in the same way.

Let the coefficient matrices G_0, G_1, G_2 and G_3 of the transfer function be given.

Determination of A and B

We now define the following augmented matrix Γ from G_0, G_1, G_2, G_3.

$$\Gamma = \begin{bmatrix} G_3 & 0 & 0 & 0 \\ G_2 & G_3 & 0 & 0 \\ G_1 & G_2 & G_3 & 0 \\ G_0 & G_1 & G_2 & G_3 \end{bmatrix} \tag{4.1}$$

Let L_3^1 be a matrix whose row vectors constitute basis vectors of row space of G_3 . Denote by T_3 any row elementary operation [10] which forms L_3^1 from G_3 and reduces zero the remaining row vectors in G_3 which are linearly dependent of L_3^1 . That is, if the rank of G_3 is k_3, then

$$T_3G_3 = \begin{bmatrix} L_3^1 \\ 0 \end{bmatrix} \begin{matrix} \} \; k_3 \\ \} \; m - k_3 \end{matrix} \tag{4.2}$$

We apply the operations T_3 to G_2, G_1, G_0 and denote the resulting matrices by

$$T_3G_2 = \begin{bmatrix} L_2^2 \\ L_2 \end{bmatrix} \qquad T_3G_1 = \begin{bmatrix} L_1^3 \\ L_{13} \end{bmatrix} \qquad T_3G_0 = \begin{bmatrix} L_0^4 \\ L_{03} \end{bmatrix}$$

If we apply the operation defined by

$$T(3) = \begin{bmatrix} T_3 & & & 0 \\ & T_3 & & \\ & & T_3 & \\ 0 & & & T_3 \end{bmatrix} \tag{4.3}$$

to the matrix Γ , then the result, denoted by Γ_3 , can be written as

$$\Gamma_3 = T(3)\Gamma = \left[\begin{array}{c|c|c|c} L_3^1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline L_2^2 & L_3^1 & 0 & 0 \\ L_2 & 0 & 0 & 0 \\ \hline L_1^3 & L_2^2 & L_3^1 & 0 \\ L_{13} & L_2 & 0 & 0 \\ \hline L_0^4 & L_1^3 & L_2^2 & L_3^1 \\ L_{03} & L_{13} & L_2 & 0 \end{array}\right] \tag{4.4}$$

Next, consider the row space of $\begin{bmatrix} L_3^1 \\ L_2 \end{bmatrix}$. Let L_2^1 be a matrix such that rows of $\begin{bmatrix} L_3^1 \\ L_2^1 \end{bmatrix}$ be basis vectors of the row space of $\begin{bmatrix} L_3^1 \\ L_2 \end{bmatrix}$. Denote by T_2 any elementary operation which forms $\begin{bmatrix} L_3^1 \\ L_2^1 \end{bmatrix}$ from $\begin{bmatrix} L_3^1 \\ L_2 \end{bmatrix}$ and reduces zero the other vectors in $\begin{bmatrix} L_3^1 \\ L_2 \end{bmatrix}$ which are linearly dependent on $\begin{bmatrix} L_3^1 \\ L_2^1 \end{bmatrix}$.

The operation T_2 is clearly of the form :

$$\begin{bmatrix} I_{k3} & 0 \\ T_{23} & T_{22} \end{bmatrix} \begin{bmatrix} L_3^1 \\ L_2 \end{bmatrix} = \begin{bmatrix} L_3^1 \\ L_2^1 \\ 0 \end{bmatrix} \tag{4.5}$$

Let k_2 be the number of row vectors contained in L_2^1. Form the augmented matrix $T(2)$ by

$$T(2) = \left[\begin{array}{cc:cc:cc:cc} \multicolumn{2}{c:}{I_m} & & & & & & \\ \hdashline 0 & 0 & I_{k3} & 0 & & & & \\ T_{23} & 0 & 0 & T_{22} & & & & \\ \hdashline & & 0 & 0 & I_{k3} & 0 & & \\ & & T_{23} & 0 & 0 & T_{22} & & \\ \hdashline & & & & 0 & 0 & I_{k3} & 0 \\ & & & & T_{23} & 0 & 0 & T_{22} \end{array}\right] \tag{4.6}$$

and compute $\Gamma_2 = T(2)\Gamma_3$.

$$\Gamma_2 = T(2)\Gamma_3 = \left[\begin{array}{c:c:c:c} L_3^1 & & & \\ 0 & & & \\ \hdashline L_2^2 & L_3^1 & & \\ L_2^1 & 0 & & \\ 0 & 0 & & \\ \hdashline L_1^3 & L_2^2 & L_3^1 & \\ L_1^2 & L_2^1 & 0 & \\ L_1 & 0 & 0 & \\ \hdashline L_0^4 & L_1^3 & L_2^2 & L_3^1 \\ L_0^3 & L_1^2 & L_2^1 & 0 \\ L_{02} & L_1 & 0 & 0 \end{array}\right] \tag{4.7}$$

Consider next the row space of $\begin{bmatrix} L_3^1 \\ L_2^1 \\ L_1 \end{bmatrix}$ and continue the operation to obtain Γ_0 finally.

$$\Gamma_0 = \left[\begin{array}{c|c|c|cccc}
L_3^1 & & & & & & \\
0 & & & & & & \\
\hline
L_2^2 & L_3^1 & & & & & \\
L_2^1 & 0 & & & & & \\
0 & 0 & & L_0^4 & L_1^3 & L_2^2 & L_3^1 \\
\hline
L_1^3 & L_2^2 & L_3^1 & L_0^3 & L_1^2 & L_2^1 & 0 \\
L_1^2 & L_2^1 & 0 & L_0^2 & L_1^1 & 0 & 0 \\
L_1^1 & 0 & 0 & L_0^1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right] \tag{4.8}$$

From the construction described above, T(3), T(2), T(1) and T(0) are nonsingular and lower-triangular block matrices. Hence we obtain the next lemma.

Lemma 3 The number k_i of row vectors in L_i^1 is equal to that of Jordan blocks of size $i + 1$, $i = o, 1, \dots, 3$.

(Proof) The proof is obvious from the independence of L_i^1 and above remarks.

The lemma 3 enables us to determine $\overline{J}$ directly from Γ_0 , i.e., $\overline{J}$ can be written down as

$$\overline{J} = \text{Diag. } \{ \overline{J}_4 \ \ \overline{J}_3 \ \ \overline{J}_2 \ \ \overline{J}_1 \} \tag{4.9}$$

where

$$\overline{J}_4 = \begin{bmatrix} 0 & I_{k_3} & 0 & 0 \\ 0 & 0 & I_{k_3} & 0 \\ 0 & 0 & 0 & I_{k_3} \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \overline{J}_3 = \begin{bmatrix} 0 & I_{k_2} & 0 \\ 0 & 0 & I_{k_2} \\ 0 & 0 & 0 \end{bmatrix}$$

$$\overline{J}_2 = \begin{bmatrix} 0 & I_{k_1} \\ 0 & 0 \end{bmatrix} \qquad \overline{J}_1 = \begin{bmatrix} 0_{k_o} \end{bmatrix}$$

We now rearrange nonzero row vectors in Γ_0 as $\overline{\Gamma}_0$ and denote $\overline{B}$ the first column block of $\overline{\Gamma}_0$.

$$\overline{\Gamma}_0 = \left[\begin{array}{cccc}
L_0^4 & L_1^3 & L_2^2 & L_3^1 \\
L_1^3 & L_2^2 & L_3^1 & \\
L_2^2 & L_3^1 & & \\
L_3^1 & & & \\
\hdashline
L_0^3 & L_1^2 & L_2^1 & \\
L_1^2 & L_2^1 & & \\
L_2^1 & & & \\
\hdashline
L_0^2 & L_1^1 & & \\
L_1^1 & & & \\
\hdashline
L_0^1 & & &
\end{array}\right] \tag{4.10}$$

$$\overline{B} = \left[\begin{array}{c}
L_0^4 \\ L_1^3 \\ L_2^2 \\ L_3^1 \\ \hdashline L_0^3 \\ L_1^2 \\ L_2^1 \\ \hdashline L_0^2 \\ L_1^1 \\ \hdashline L_0^1
\end{array}\right] \tag{4.11}$$

Lemma 4   The row vectors in $\bar{\Gamma}_0$ are linearly independent.

Lemma 5   The $\{ \bar{J}, \bar{B} \}$ defined by (4.9) and (4.11) is a controllable pair.
(Proof) With the use of $\bar{J}$ and $\bar{B}$, $\bar{\Gamma}_0$ can be written as

$$\bar{\Gamma}_0 = [\ \bar{B} \ \vdots \ \bar{J}\bar{B} \ \vdots \ \bar{J}^2\bar{B} \ \vdots \ \bar{J}^3\bar{B}\ ] \tag{4.12}$$

Taking account of the fact that $\bar{J}^k = 0$ for $k \geq 4$, it is obvious that the rank of $\bar{\Gamma}_0$ is equal to that of the controllability matrix of $\{\bar{J}, \bar{B}\}$. In view of the proceeding lemma $\{\bar{J}, \bar{B}\}$ is thus proved to be controllable.

Determination of C matrix

Consider the linear matrix equation in M :

$$M\bar{\Gamma}_0 = [G_0 \quad G_1 \quad G_2 \quad G_3] \tag{4.13}$$

Since

$$\mathrm{rank}\begin{bmatrix} & \bar{\Gamma}_0 & & \\ \hline G_0 & G_1 & G_2 & G_3 \end{bmatrix} = \mathrm{rank}\ \bar{\Gamma}_0 \ ,$$

(4.13) has a solution. Moreover, as row vectors in $\bar{\Gamma}_0$ are linearly independent, M exists uniquely. To be more specific, the i-th row of M is the representation of i-th row of $[G_0 \quad G_1 \quad G_2 \quad G_3]$ with respect to basis vectors of $\bar{\Gamma}_0$. Concerning this solution, we can give a system theoretic statement.

Lemma 4   The $\{\bar{J}, \bar{C}\}$ is an observable pair, where $\bar{C}$ is the unique solution of (4.13).
Proof : In view of (4.13), the unique solution $\bar{C}$ satisfies $\bar{C}\,\bar{B} = G_0$, $\bar{C}\,\bar{J}\,\bar{B} = G_1$, $\bar{C}\,\bar{J}^2\bar{B} = G_2$ and $\bar{C}\,\bar{J}^3\bar{B} = G_3$. As $\bar{J}^k = 0$ for $k \geq 4$,

$$\begin{bmatrix} \bar{C}\,\bar{J}^3 \\ \vdots \\ \bar{C} \end{bmatrix} \bar{\Gamma}_0 = \Gamma$$

Since

$$\mathrm{rank}\ \Gamma = \mathrm{rank}\ \bar{\Gamma}_0 = n_0 \ ,$$

the lemma follows from the Sylvester's Theorem for the rank of product of rectangular matrix [10].

We summarize the above results in the next.

Theorem 2   The irreducible Jordan canonical realization is obtained in the following steps :

(1) Form the augmented matrix Γ and compute $\bar{\Gamma}_0$.
(2) $\bar{J}$ is determined from the numbers k_i of the row vectors in L_i^1 .
(3) Let $\bar{B}$ the first column block of $\bar{\Gamma}_0$.
(4) Let $\bar{C}$ the solution of $M\bar{\Gamma}_0 = [G_0\ G_1\ G_2\ G_3]$.
(5) Interchange rows and columns of $\{ \bar{J}, \bar{B}, \bar{C} \}$ so that $\bar{J}$ be transformed into J .

Then $\{\lambda In + J, B, C\}$ is the desired irreducible Jordan form realization of G(s) .

Remark 1 : If we choose the linearly independent column vectors from Γ in the anologous way, we can also obtain another irreducible Jordan form realization. In this case, $\bar{J}$ and $\bar{C}$ are determined in the first place and then $\bar{B}$ is computed.

Remark 2 : The process of transforming $\{\bar{J}, \bar{B}, \bar{C}\}$ into $\{J, B, C\}$ can be carried out easily on machine by forming a composite matrix $\begin{bmatrix} \bar{J} & \bar{B} \\ \bar{C} & 0 \end{bmatrix}$ and interchanging rows and columns of the composite matrix so that $\bar{J}$ becomes J .

5. EXAMPLE

Now we illustrate the proposed algorithm by realizing the following rational matrix treated by many authors [4] ∿ [6] .

$$G(s) = \frac{1}{s^4}\begin{bmatrix} s^3-s^2+1 & 1 & -s^3+s^2-2 \\ 1.5s+1 & s+1 & -1.5s-2 \\ s^3-9s^2-s+1 & -s^2+1 & s^2-s-2 \end{bmatrix}$$

Then coefficients matrices are given by

$$G_0=\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad G_1=\begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ -9 & -1 & 0 \end{bmatrix} \quad G_2=\begin{bmatrix} 0 & 0 & 0 \\ 1.5 & 1 & -1.5 \\ -1 & 0 & -1 \end{bmatrix} \quad G_3=\begin{bmatrix} 1 & 1 & -2 \\ 1 & 1 & -2 \\ 1 & 1 & -2 \end{bmatrix}$$

Take $T_3=\begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$, then

$$\Gamma_3=T(3)\Gamma=\left[\begin{array}{ccc|ccc|ccc|ccc} 1 & 1 & -2 & & & & & & & & & \\ 0 & 0 & 0 & & & & & & & & & \\ 0 & 0 & 0 & & & & & & & & & \\ \hline 0 & 0 & 0 & 1 & 1 & -2 & & & & & & \\ 1.5 & 1 & -1.5 & 0 & 0 & 0 & & & & & & \\ -1 & 0 & -1 & 0 & 0 & 0 & & & & & & \\ \hline -1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & -2 & & & \\ 1 & 0 & -1 & 1.5 & 1 & -1.5 & & & & & & \\ -8 & 0 & -1 & -1 & 0 & -1 & & 0 & & & & \\ \hline 1 & 0 & -1 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & -2 \\ -1 & 0 & 1 & 1 & 0 & -1 & 1.5 & 1 & -1.5 & & & \\ 0 & 0 & 2 & -8 & 0 & -1 & -1 & 0 & -1 & & 0 & \end{array}\right]$$

As $\begin{bmatrix} L_3^1 \\ L_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -2 \\ 1.5 & 1 & -1.5 \\ -1 & 0 & -1 \end{bmatrix}$, T_2 can be taken as $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 2 & -2 & -1 \end{bmatrix}$. Continuing the process to obtain Γ_0 , we finally arrive at $\bar{\Gamma}_0$.

$$\bar{\Gamma}_0 = \left[\begin{array}{ccc|ccc|ccc|ccc} 1 & 0 & -1 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & -2 \\ -1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & -2 & & & \\ 0 & 0 & 0 & 1 & 1 & -2 & & & & & & \\ 1 & 1 & -2 & & & & & & & & & \\ \hline -1 & 0 & 3 & -8 & -1 & -1 & 0 & 1 & -3 & & & \\ -8 & -1 & -1 & 0 & 1 & -3 & & & & & & \\ 0 & 1 & -3 & & & & & & & & & \\ \hline -40 & -5 & -7 & & & & & & & & & \end{array}\right]$$

From this it is clear that $k_0=1$, $k_1=0$, $k_2=1$ and $k_3=1$, i.e., the number of Jordan block of size 1, 3 and 4 is one, respectively. The $\bar{\Gamma}_0$ also gives $\bar{B}$.

Thus

$$\overline{J} = \begin{bmatrix} 0 & 1 & 0 & 0 & & & & \\ 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 1 & & & & \\ 0 & 0 & 0 & 0 & & & & \\ & & & & 0 & 1 & 0 & \\ & & & & 0 & 0 & 1 & \\ & & & & 0 & 0 & 0 & \\ & & & & & & & 0 \end{bmatrix} \quad \text{and} \quad \overline{B} = \left[\begin{array}{rrr} 1 & 0 & -1 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & -2 \\ \hdashline -1 & 0 & 3 \\ -8 & -1 & -1 \\ 0 & 1 & -3 \\ \hdashline -40 & -5 & -7 \end{array}\right]$$

We next solve the following equation

$$M\,\overline{\Gamma}_0 = \left[\begin{array}{rrrrrrrrrrrr} 1 & 0 & -1 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.5 & 1 & -1.5 & 1 & 1 & -2 \\ 1 & 0 & 1 & -9 & -1 & 0 & -1 & 0 & -1 & 1 & 1 & -2 \end{array}\right]$$

and obtain a unique solution

$$\overline{C} = \left[\begin{array}{rrrrrrrr} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1.5 & -3 & 0 & -0.5 & 2.5 & 0 & -0.5 \\ 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right]$$

In this example the fifth step of Theorem 2 is unnecessary. The $\{\overline{J}, \overline{B}, \overline{C}\}$ itself is the desired irreducible Jordan canonical form realization.

6. CONCLUSION

A new and simple procedure for realizing an irreducible Jordan canonical form of a rational transfer matrix is proposed. The method is straight forward and is convenient for both machine and hand computations.

References

[1] R.E. Kalman : Mathematical description of linear dynamical systems, SIAM J. Control vol 1 (1963)

[2] E.G. Gilbert : Controllability and observability in multivariable control systems, ibid.

[3] L.A. Zadeh and C.A. Desoer : Linear System Theory McGraw Hill (1963)

[4] R.E. Kalman : Irreducible realizations and degree of a rational matrix, J. SIAM vol 13 (1965)

[5] S.P. Panda and C.T. Chen : Irreducible Jordan form realization of a rational matrix, IEEE AC-14 (1969)

[6] Yen-Long Kuo : On the irreducible Jordan form realization and the degree of a rational matrix, IEEE CT-17 (1970)

[7] M. Lal, S.C. Puri and H. Sigh : On the realization of a linear time-invariant dynamical systems, IEEE AC-17 (1972)

[8] H.L. Huang : A generalized-Jordan-form-approach one-step irreducible realization of matrices, IEEE AC-19 (1974)

[9] C.T. Chen and C.A. Desoer : A proof of controllability of Jordan form state equations, IEEE AC-13 (1968)

[10] F.R. Gantmacher : The Theory of Matrices, vol 1. Chelsea (1959)

NEW PROBLEMS IN PARTIAL REALIZATION THEORY

JAIME R. ROMAN
The Analytic Sciences Corporation
6 Jacob Way
Reading, Massachusetts 01867

ABSTRACT

The structure of non-minimal partial realizations is studied, and three new problems in partial realization theory are discussed. These are: 1) partial realizations of incompletely-specified finite matrix sequences, 2) the minimal partial stable realization problem, and 3) the minimal partial realization with arbitrary poles problem.

INTRODUCTION

Consider a system described by the equations

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t) \qquad (1a)$$

$$\underline{y}(t) = C\,\underline{x}(t) \qquad (1b)$$

where $\underline{x}(t)$ is the n-dimensional state vector, $\underline{u}(t)$ is the m-dimensional input vector, $\underline{y}(t)$ is the r-dimensional output vector, and A,B,C are real, constant matrices of appropriate dimensions. System (1) is also denoted here by the matrix triple $(A,B,C)_n$, and is assumed to be completely controllable and observable. It is also assumed (for notational simplicity) that $m < n$, $r < n$, and B and C are full-rank matrices. It is well known [1] that the impulse response matrix of system (1) is

$$H(t) = \sum_{i=1}^{\infty} H_i \frac{t^{i-1}}{(i-1)!} \qquad (2)$$

where

$$H_i = CA^{i-1}B \qquad (3)$$

The complete realization problem is defined as follows. Given an infinite sequence $\{H_i : i \geq 1\}$ of r×m matrices find a triple $(A,B,C)_n$ such that (3) is satisfied for $i \geq 1$. The partial realization problem is defined as: given a finite sequence $\{H_i : 1 \leq i \leq N\}$ of r×m matrices find a triple $(A,B,C)_n$ such that (3) is satisfied for $1 \leq i \leq N$. In each case the triple $(A,B,C)_n$ is said to be a realization of the given sequence. These two problems have been studied considerably and there are several different solutions available in the literature. However, past research efforts have been directed mostly towards the case where the dimension of the realization $(A,B,C)_n$ is to be minimal [1] - [6].

In this paper the emphasis is on non-minimal partial realizations. The structure of non-minimal partial realizations is studied, and a procedure is given to construct all non-minimal partial realizations of a given dimension. Three new problems in partial realization theory are briefly discussed. These are: 1) partial realizations of incompletely-specified finite matrix sequences, 2) the minimal partial stable realization problem, and 3) the minimal partial realization with arbitrary poles problem. These problems have been shown to arise in the design of observers [7], [10].

It is well known [12] that a completely observable system (1) can always be represented in the form

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1r} \\ \vdots & & \vdots \\ A_{r1} & \cdots & A_{rr} \end{bmatrix} \tag{4a}$$

$$A_{ii} = \begin{bmatrix} \underline{0}_{\nu_i - 1} & I_{\nu_i - 1} \\ \multicolumn{2}{c}{-\underline{\theta}^T_{ii}} \end{bmatrix} \qquad A_{ij} = \begin{bmatrix} 0_{\nu_i - 1, \nu_j} \\ -\underline{\theta}^T_{ij} \end{bmatrix} \tag{4b}$$

$$\underline{\theta}^T_{ii} = \begin{bmatrix} \theta_{ii1} & \theta_{ii2} & \cdots & \theta_{ii\nu_i} \end{bmatrix} \tag{4c}$$

$$\underline{\theta}^T_{ij} = \begin{bmatrix} \theta_{ij1} & \theta_{ij2} & \cdots & \theta_{ij\kappa_{ij}} & 0 & \cdots & 0 \end{bmatrix} \tag{4d}$$

$$B = \begin{bmatrix} B_1 \\ \vdots \\ B_r \end{bmatrix} \qquad B_i = \begin{bmatrix} \underline{b}^T_{i1} \\ \vdots \\ \underline{b}^T_{i\nu_i} \end{bmatrix} \tag{5}$$

$$C = \begin{bmatrix} \underline{c}^T_1 \\ \vdots \\ \underline{c}^T_r \end{bmatrix} = \begin{bmatrix} \underline{i}^T_1 \\ \vdots \\ \underline{i}^T_{n-\nu_r+1} \end{bmatrix} \tag{6}$$

where I_j is the j×j identity matrix and $\underline{i}_k$ is its kth column, $0_{j,k}$ is a j×k null matrix and $\underline{0}_j$ is one of its columns, and the superscript T indicates vector (or matrix) transposition. Also, $\{\nu_i : 1 \leq i \leq r\}$ is the (ordered) set of observability indices of the pair $(A,C)_n$, and $\{\kappa_{ij} : 1 \leq i,j \leq r\}$ and $\{\theta_{ijk} : 1 \leq i,j \leq r; 1 \leq k \leq \kappa_{ij}\}$ are a set of integers and a set of real numbers, respectively, defined by the relations

$$\underline{c}^T_i A^{\nu_i + \tau} B = -\sum_{j=1}^{r} \sum_{k=1}^{\kappa_{ij}} \theta_{ijk}\, \underline{c}^T_j A^{k-1+\tau} B \qquad \begin{matrix} i=1,2,\ldots,r \\ \tau=0,1,2,\ldots \end{matrix} \tag{7}$$

$$\kappa_{ij} = \begin{cases} \nu_i + 1 & \text{if} \quad \nu_i < \nu_j \quad \text{and} \quad i > j \\ \nu_i & \text{if} \quad \nu_i \leq \nu_j \quad \text{and} \quad i \geq j \\ \nu_j & \text{if} \quad \nu_i \geq \nu_j \end{cases} \tag{8}$$

If the pair $(A,B)_n$ is completely controllable then the set $\{\theta_{ijk}\}$ (the enumeration of the subscripts is dropped in the sequel) is unique and the form (4) - (6) is a canonical form for the set of all controllable and observable triples $(A,B,C)_n$ under similarity transformations in the state-space [7] (the sets $\{\nu_i\}$, $\{\theta_{ijk}\}$, and the rows of B are system invariants). It follows from (3) and (7) that

$$\underline{h}^T_{\nu_i+1+\tau,i} = -\sum_{j=1}^{r}\sum_{k=1}^{\kappa_{ij}} \theta_{ijk}\,\underline{h}^T_{k+\tau,j} \qquad \begin{array}{l} i=1,2,\ldots,r \\ \tau=0,1,2,\ldots \end{array} \tag{9}$$

where $\underline{h}^T_{pq}$ is the q^{th} row of H_p.

Consider a finite matrix sequence $\{H_i : 1 \le i \le N\}$. It has been shown elsewhere [7] that the (ordered) set $\{\nu_i\}$ associated with all minimal realizations of this sequence is unique, the set $\{\theta_{ijk}\}$ is generally non-unique, and the number of different minimal realizations (in the coordinate-free sense) of the given finite sequence is equal to the number of different sets $\{\theta_{ijk}\}$ for which (9) is satisfied (for $\tau \le N+1-\max(\nu_i)$).

Let n_M denote the dimension of a minimal realization of the given partial sequence and let $\{\nu_{iM}\}$ denote the unique, ordered set of minimal observability indices (observability indices of a minimal realization). An ordered set of indices $\{\nu_i : 1 \le i \le r\}$ is called an admissible set of observability indices of order n for the finite sequence $\{H_i\}$ if:

$$\nu_1 + \nu_2 + \ldots + \nu_r = n > n_M \tag{10a}$$

$$\nu_i \ge \nu_{iM} \qquad i=1,2,\ldots,r \tag{10b}$$

Further define $N(n)$ as the class of all sets of admissible observability indices of order n. Note that $N(n_M)$ consists of only the set $\{\nu_{iM}\}$, and $N(n)$ is the empty set if $n < n_M$.

Suppose that it is desired to construct a non-minimal controllable and observable realization of dimension n of the given finite sequence. It follows from the above discussion and definitions that such a realization, if it exists, admits a representation of the form (2.1) - (2.3) and has an output (or observability) structure $\{\nu_i\} \subset N(n)$. The existence (or non-existence) of a controllable and observable non-minimal partial realization of dimension n seems to be difficult to show in a direct manner; it is checked here indirectly via the procedure given below for the construction of non-minimal partial realizations of dimension n.

Before describing the construction procedure it is convenient to define the following Hankel-type array associated with a partial sequence $\{H_i : 1 \le i \le N\}$:

$$\mathcal{H}_{ijk} = \begin{bmatrix} \underline{h}^T_{1k} & \underline{h}^T_{2k} & \cdots & \underline{h}^T_{jk} \\ \underline{h}^T_{2k} & \underline{h}^T_{3k} & \cdots & \underline{h}^T_{j+1,k} \\ \vdots & \vdots & & \vdots \\ \underline{h}^T_{ik} & \underline{h}^T_{i+1,k} & \cdots & \underline{h}^T_{i+j-1,k} \end{bmatrix} \tag{11}$$

(this array is not the standard Hankel array used in realization theory) with $\underline{h}_{pq}^T$ as previously defined. The construction procedure presented below gives all non-minimal controllable and observable partial realizations of dimension n (it can be easily modified to give only one or several partial realizations of dimension n). The procedure assumes as an input the set $\mathcal{N}(n)$ (recall that $\mathcal{N}(n)$ is easily obtained given the set $\{\nu_{iM}\}$, which, in turn, is computed in a straightforward manner [8]).

NON-MINIMAL PARTIAL REALIZATION CONSTRUCTION

Step 1. Select a set $\{\nu_i\} \subset \mathcal{N}(n)$.

Step 2. Compute the corresponding parameters $\{\theta_{ijk}\}$ by solving the following equation:

$$-\begin{bmatrix}\underline{\theta}_{i1}^T & \cdots & \underline{\theta}_{ir}^T\end{bmatrix}\begin{bmatrix}\mathcal{H}_{\nu_1,N-\nu_i,1}\\ \vdots \\ \mathcal{H}_{\nu_r,N-\nu_i,r}\end{bmatrix} = \begin{bmatrix}\underline{h}_{\nu_i+1,i}^T & \cdots & \underline{h}_{N,i}^T\end{bmatrix}$$

for i=1,2, . . . , r. This can be written more compactly as

$$-\underline{\theta}_i^T\tilde{\mathcal{H}}_i = \underline{\tilde{h}}_i^T \qquad i=1,2,\ldots,r \tag{12}$$

Equation (12) generally admits many solutions $\underline{\theta}_i$, and there is always at least one solution. In the cases where $\nu_i \geq N$, matrix $\tilde{\mathcal{H}}_i$ is undefined; this results in $\underline{\theta}_i$ being arbitrary.

Step 3. With the sets $\{\nu_i\}$ and $\{\theta_{ijk}\}$ (the set $\{\theta_{ijk}\}$ may include several arbitrary parameters) construct the pair $(A,C)_n$ in the form (4), (6). The rows of B are obtained as

$$\underline{b}_{ij}^T = \underline{h}_{ji}^T \qquad i,j=1,2,\ldots,r \tag{13}$$

Step 4. Check the controllability of the pair $(A,B)_n$ constructed in Step 3.

Step 5. Repeat Steps 2-4 with a different set $\{\nu_i\} \subset \mathcal{N}(n)$. This is continued until all the elements of $\mathcal{N}(n)$ are considered.

The above procedure is related to those in [8].[1] One pass through the procedure gives all non-minimal partial realizations of dimension n that have the same observability structure, while every allowable choice of the arbitrary parameters in the set $\{\theta_{ijk}\}$ gives a different realization. The controllability requirement determines which values are allowable for the arbitrary θ_{ijk}. In fact, it is possible that there do not exist controllable and observable partial realizations of a specified dimension $n > n_M$ (this important observation does not seem to have been noted elsewhere). This is illustrated in the following example.

EXAMPLE 1. Consider the sequence $H_1=I_2$, $H_2=0_{2,2}$, $H_3=0_{2,2}$. Following [8] it is easy to verify that $n_M=2$, $\nu_{1M}=\nu_{2M}=1$, and the unique minimal realization is $\{A,B,C\}_2 = \{0_{2,2},I_2,I_2\}$. $\mathcal{N}(3)$ consists of {2,1} and {1,2}. Using the construction procedure given here it is found that all partial realizations of dimension equal to 3 are represented by the following two matrix triples:

---

[1] The discussion in [8] and [10] is based on the dual point of view.

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\theta_{112} & 0 \\ 0 & -\theta_{212} & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

with θ_{112} and θ_{212} arbitrary, and

$$A - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\theta_{222} \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

with θ_{222} arbitrary. Notice that both of these triples are uncontrollable for all values of the arbitrary parameters; however, controllable and observable partial realizations of dimension 4 do exist.

PARTIAL REALIZATIONS OF INCOMPLETELY-SPECIFIED SEQUENCES

A finite matrix sequence which has one or more of its matrix elements with some unspecified columns is defined here as an incompletely-specified partial sequence. The incompletely-specified sequences considered in this paper are limited to those which satisfy the following rule: if the j^{th} column of H_i is unspecified, then the j^{th} columns of $H_{i+1}, \ldots, H_N$ are also unspecified. This type of incompletely-specified sequence arises when an observer design problem is represented as a partial realization problem [7], [10].

It has been shown elsewhere [7] that all minimal realizations of a finite matrix sequence have the same sets of controllability and observability indices. However, this is not true in general for minimal realizations of a finite incompletely-specified matrix sequence. Incompletely-specified matrix sequences of the type considered here have a unique set of minimal observability indices and, in general, a non-unique set of minimal controllability indices.

This non-uniqueness is not easily handled in a very general way with most partial realization algorithms. But if (the dual of) Algorithm 1 in [8] (slightly modified) is used to obtain all minimal partial realizations, and if the realization procedure of the preceding section (slightly modified) is used to compute all non-minimal partial realizations of dimension n, then these problems are actually simpler than in the completely-specified sequence case. These ideas are probably described best using an example.

EXAMPLE 2. Consider the sequence $H_1=[I_2\ \underline{0}_3]$, $H_2=[*\ \underline{i}_1\ \underline{0}_2]$, $H_3=[*\ \underline{i}_2\ *]$ of 2×3 matrices (the asterisks, *, represent 2-dimensional vectors of unspecified elements). Applying (the dual of) Algorithm 1 in [8] to this sequence (with the provision that the unspecified elements are not considered in examining the rows of the standard Hankel array associated with the sequence), it is found that $n_M=2$, $\nu_{1M}=\nu_{2M}=1$, and the unique minimal realization is: $A=[\underline{i}_2\ \underline{i}_1]$, $B=[I_2\ \underline{0}_2]$, $C=I_2$. $\mathcal{N}(3)$ consists of {2,1} and {1,2}, but only the structure {1,2} is considered here, for brevity. The relations (12) are

$$-\begin{bmatrix} \theta_{111} & \theta_{121} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & * & 1 & 0 \\ 0 & 1 & 0 & * & 0 & 0 \\ * & 0 & 0 & * & 1 & * \end{bmatrix} = \begin{bmatrix} * & 1 & 0 & * & 0 & * \end{bmatrix}$$

$$-\begin{bmatrix} \theta_{211} & \theta_{221} & -\theta_{222} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ * & 0 & 0 \end{bmatrix} = \begin{bmatrix} * & 1 & * \end{bmatrix}$$

Notice that the columns of $\tilde{\mathcal{H}}_i$ $(i=1,2)$ which correspond to columns of $\tilde{\underline{h}}_i^T$ which contain an unspecified element can be deleted because there are no constraints on the values of the unspecified elements. This reduces the number of equations that must be solved (from 9 to 4). The triple thus obtained is:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\theta_{211} & 1 & -\theta_{222} \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ b_{221} & 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

with θ_{211}, θ_{222}, and b_{221} completely arbitrary. The pair $(A,B)_3$ is controllable if and only if either $b_{221} \neq 1$ or $\theta_{222}\, b_{221} \neq -\theta_{211}$.

This example suggests the modification to the procedure of the preceding section to handle incompletely-specified matrix sequences; it is as follows. In Step 2: before solving (12) for $\underline{\theta}_i$, delete the columns of $\tilde{\mathcal{H}}_i$ which correspond to columns of $\tilde{\underline{h}}_i^T$ which are unspecified. The corresponding modification to Algorithm 1 in [8] is similar.

MINIMAL PARTIAL STABLE AND ARBITRARY REALIZATIONS

In some applications [10] it is often required to realize a finite matrix sequence with: 1) a stable matrix triple of dimension as low as possible, or 2) a matrix triple with specified poles and of dimension as low as possible. These problems are referred to here as the minimal partial stable realization problem and the minimal partial arbitrary realization problem, respectively.

Consider first the minimal partial arbitrary realization problem. Let n_A denote the dimension of a minimal partial arbitrary realization, and let

$$\kappa = \sum_{i=1}^{r} \sum_{j=1}^{r} \kappa_{ij} \tag{14}$$

where the integers κ_{ij} are computed according to (8) for a set $\{\nu_i\} \subset N(n)$. The integer κ is the number of $\{\theta_{ijk}\}$ parameters in the canonical form (4)-(6). Two important results are now stated (see [7] for the proofs).

RESULT 1. For a triple $(A,B,C)_n$ to be a partial arbitrary realization of a finite matrix sequence $\{H_i\}$ it is necessary that

a) its set of observability indices be an admissible set, and

b) $n \leq \kappa - \sum_{i=1}^{r} \rho(\overline{\mathcal{H}}_i)$

where $\rho(\cdot)$ denotes "rank of", and $\overline{\mathcal{H}}_i$ is matrix $\tilde{\mathcal{H}}_i$ with the rows which correspond to zero elements in $\underline{\theta}_i$ deleted.

RESULT 2. A triple $(A,B,C)_n$ with

a) $n \geq \min(rN,mN)$, and

b) observability indices $\{\nu_i\} \subset N(n)$

can always be constructed to represent a partial arbitrary realization of a finite matrix sequence $\{H_i : 1 \leq i \leq N\}$.

RESULT 1 gives necessary conditions, while RESULT 2 gives sufficient conditions for a triple to be a partial arbitrary realization. RESULT 2 also implies that a minimal partial arbitrary realization always exists. These two results bound the value of n_A corresponding to a given sequence. It appears to be difficult to obtain a stronger result, except for vector (rather than matrix) sequences in which cases it has been shown [7] that $n_A = N$.

Consider now the minimal partial stable realization problem. Let n_S denote the dimension of a minimal partial stable realization. The minimal partial stable realization is generally non-unique, and it follows from RESULT 2 that one exists for every finite matrix sequence. It is also true that

$$n_A \geq n_S \geq n_M \tag{15}$$

The value of n_M and the set $\{\nu_{iM}\}$ are obtained fairly easily, and so are also the bounds for n_A. Unfortunately, it appears to be difficult to obtain n_S (or strong bounds for it) without actually computing realizations and checking stability. In many cases, though, the relation (15) provides strong bounds for n_S More complicated (but not much stronger) bounds are given in [8] and [9].

Minimal partial stable realization algorithms are given in [8] and [9]. The algorithm in [8] is based on the structure approach discussed here and uses the canonical form (4) - (6). Other partial realization algorithms can be used to obtain minimal partial stable realizations, but usually at the expense of more computations and/or some loss of generality (for example, the partial realization algorithm in [5] uses a different canonical form which is more convenient for checking stability, but it has been shown [7], [11] that the resulting realizations are not always minimal). The minimal partial stable realization algorithm in [9] is considerably different from the one in [8]; it consists of investigating the rank of various submatrices of the standard Hankel array associated with the partial sequence.

The non-minimal realization procedure given here can be used to search for all minimal partial arbitrary realizations by noticing that RESULT 1 determines a lower bound for n_A and also determines which admissible sets of observability indices are candidates for the output structure of a minimal partial arbitrary realization. The resulting algorithm is similar to Algorithm 2 in [8].

EXAMPLE 3. Consider the sequence

$$H_1 = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \end{bmatrix} \quad H_2 = \begin{bmatrix} 2 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad H_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Using Algorithm 1 in [8] it is easy to verify that $n_M = 6$ and $\nu_{1M} = \nu_{2M} = 3$. Notice that $rN = (2)(3) = 6$; thus, it follows from RESULT 2 that $n_A = n_S = n_M$ for this problem. The set $N(6)$ consists of only the element $\{3,3\}$; and since $\nu_1 = \nu_2 = N$, it follows that $\underline{\theta}_1$ and $\underline{\theta}_2$ in (12) are arbitrary vectors (see Step 2). All minimal partial realizations of the given sequence are represented by the triple

$$A = \begin{bmatrix} \underline{i}_2^T \\ \underline{i}_3^T \\ -\underline{\theta}_1^T \\ \underline{i}_5^T \\ \underline{i}_6^T \\ -\underline{\theta}_2^T \end{bmatrix} \quad B = \begin{bmatrix} \underline{h}_{11}^T \\ \underline{h}_{21}^T \\ \underline{h}_{31}^T \\ \underline{h}_{12}^T \\ \underline{h}_{22}^T \\ \underline{h}_{32}^T \end{bmatrix} \quad C = \begin{bmatrix} \underline{i}_1^T \\ \underline{i}_4^T \end{bmatrix}$$

with

$$\underline{\theta}_1^T = \begin{bmatrix} \theta_{111} & \theta_{112} & \theta_{113} & \theta_{121} & \theta_{122} & \theta_{123} \end{bmatrix}$$

$$\underline{\theta}_2^T = \begin{bmatrix} \theta_{211} & \theta_{212} & \theta_{213} & \theta_{221} & \theta_{222} & \theta_{223} \end{bmatrix}$$

two completely arbitrary vectors.

CONCLUSIONS

The existence of non-minimal partial realizations and their output structure is studied. Based on this a procedure is given to construct all non-minimal partial realizations of a given dimension. The realizations are constructed in a canonical form, which greatly simplifies computations. The construction procedure is applied to incompletely-specified finite matrix sequences. The minimal partial stable realization problem and the minimal partial arbitrary realization problem are briefly considered, and several results are stated. Further research is required on these problems, especially the minimal partial stable realization problem since the available results are rather weak.

REFERENCES

[1] B. L. Ho and R. E. Kalman,"Effective Construction of Linear State Variable Models From Input/Output Functions," Regelungstechnik, Vol. 14, No. 12, pp. 545-548, January 1966.

[2] R. E. Kalman, P. L. Falb, and M. A. Arbib, Topics in Mathematical System Theory, McGraw-Hill, New York, 1969.

[3] A. J. Tether, "Construction of Minimal Linear State-Variable Models From Finite Input-Output Data," IEEE Transactions on Automatic Control, Vol. AC-15, No. 4, pp. 427-436, August 1970.

[4] R. E. Kalman, "On Minimal Partial Realizations of a Linear Input/Output Map," Aspects of Network and System Theory, R. E. Kalman and N. DeClaris, eds., pp. 385-407, Holt-Rinehart-Winston, New York, 1971.

[5] J. E. Ackermann, "On Partial Realizations," IEEE Transactions on Automatic Control (Correspondence), Vol. AC-17, No. 3, pg. 381, June 1972.

[6] B. W. Dickinson, M. Morf, and T. Kailath, "A Minimal Realization Algorithm for Matrix Sequences," IEEE Transactions on Automatic Control, Vol. AC-19, No. 1, pp. 31-38, February 1974.

[7] J. R. Roman, "Low Order Observer Design via Realization Theory," Ph.D. dissertation, Department of Electrical Engineering, University of Florida, Gainesville, Florida, June 1975.

[8] J. R. Roman and T. E. Bullock, "Minimal Partial Realizations in a Canonical Form," IEEE Transactions on Automatic Control (Short Papers), Vol. AC-20, No. 4, pp. 529-530, August 1975.

[9] J. B. Moore and G. Ledwich, "Minimal Stable Partial Realizations," Preprints IFAC 75,Preprint Part IB, Cambridge, Massachusetts, August 1975.

[10] J. R. Roman and T. E. Bullock, "Design of Minimal Order Stable Observers for Linear Functions of the State Via Realization Theory," IEEE Transactions on Automatic Control, Vol. AC-20, No. 5, pp.613-622, October 1975.

[11] G. Ledwich and T. E. Fortmann, "Comments on 'On Partial Realizations'," IEEE Transactions on Automatic Control (Correspondence), Vol. AC-19, No. 5, pp. 625-626, October 1974.

[12] D. G. Luenberger, "Canonical Forms for Linear Multivariable Systems," IEEE Transactions on Automatic Control (Short Papers), Vol. AC-12, No. 3, pp. 290-293, June 1967.

ON TIME DOMAIN SENSITIVITY OF LINEAR SYSTEMS

MANSOUR ESLAMI and DAVID P. BROWN
Department of Electrical and Computer Engineering
University of Wisconsin - Madison
Madison, Wisconsin 53706

ABSTRACT

Some formulas for the sensitivity of the state vector and two sensitivity invariants for a linear, time-invariant system are established. Zero sensitivity is related to observability and controllability.

I. INTRODUCTION

Many researchers have worked on problems in the areas of network and system sensitivity, however much remains to be done. Most of the work in the linear, time-invariant network and systems areas has been done using transfer function methods, for example [1,2]. It is of interest to use state-variable methods [3,4] to establish network sensitivity properties because of the generality and additional insight gained.

In this paper state-variable methods are used to establish some new properties of linear time-invariant systems or networks. Some sensitivity formulas are established (Theorems 1, 2 and 3) which permit the evaluation of sensitivity functions. Two new time domain sensitivity invariants are established (Theorems 4 and 5) and a simple relation between zero sensitivity and observability and controllability is given (Theorem 6).

II. SOME SENSITIVITY FORMULAS

In the following theorem an expression for the change in the state vector of a linear system in terms of changes in the system parameter matrices is presented.

Theorem 1

Consider the system

$$\dot{X}(t) = AX(t) + BU(t), \qquad X(o) = X_o$$

where $X(t)$ is an $n\times 1$ state vector, $U(t)$ is an $r\times 1$ input vector and A,B are, respectively, $n\times n$, $n\times r$, constant matrices. Suppose $A\rightarrow A+\Delta A$, $B\rightarrow B+\Delta B$ and $X(t)\rightarrow X(t)+\Delta X(t)$ where $X(o)+\Delta X(o) = X_o$, then

$$\Delta X(t) = \int_o^t [\exp(A+\Delta A)(t-s)][\Delta AX(s)+\Delta BU(s)]ds \ .$$

Proof: By hypothesis it follows that

$$\dot{X}(t) + \Delta\dot{X}(t) = (A+\Delta A)(X+\Delta X) + (B+\Delta B)\ U(t)$$

where $X(o) + \Delta X(o) = X_o$.

This is easily simplified to

$$\Delta\dot{X}(t) = (A+\Delta A)\ \Delta X(t) + \Delta AX(t) + \Delta BU(t)$$

which has the following solution;

$$\Delta X(t) = [\exp(A+\Delta A)(t)]\Delta X(o) + \int_o^t [\exp(A+\Delta A)(t-s)][\Delta AX(s)+\Delta BU(s)]ds \ .$$

Since $\Delta X(o) = 0$ by hypothesis, the expression given in the conclusion follows.

It is noted that the conclusion of Theorem 1 is valid for small or large variations in the system parameter matrices.

An expression for the sum of the sensitivity functions of the state variables of a linear system is presented in the following theorem.

Theorem 2

Consider the state equations of a linear system,

$$\dot{X}(t) = AX(t) + BU(t), \qquad X(o) = X_o$$

where $A = [a_{ij}]$ and $B = [b_{ij}]$. The sum of the (unnormalized) sensitivity functions of the system always satisfy,

$$\sum_{i,j} \alpha_{ij} \frac{\partial X(t)}{\partial \alpha_{ij}} = \int_o^t [\exp(A)\ (t-s)][AX(s) + BU(s)]ds$$

where $\alpha_{ij} = a_{ij}, b_{ij}$ for all i,j.

Proof: Suppose ΔA has only one nonzero entry Δa_{ij}, that is

$$\Delta A = \begin{bmatrix} & 0 & \\ 0 & \Delta a_{ij} & 0 \\ & 0 & \end{bmatrix}$$

and $\Delta B = 0$. Then by Theorem 1,

$$\frac{\Delta X(t)}{\Delta a_{ij}} = \int_o^t [\exp(A+\Delta A)(t-s)] \begin{bmatrix} & 0 & \\ 0 & 1_{ij} & 0 \\ & 0 & \end{bmatrix} X(s)\ ds$$

and taking the limit as $\Delta a_{ij} \to 0$ results in

$$\frac{\partial X(t)}{\partial a_{ij}} = \int_o^t [\exp(A)(t-s)] \begin{bmatrix} & 0 & \\ 0 & 1_{ij} & 0 \\ & 0 & \end{bmatrix} X(s)\ ds \quad .$$

It therefore follows that

$$\sum_{i,j} a_{ij} \frac{\partial X(t)}{\partial a_{ij}} = \int_o^t [\exp(A)(t-s)]AX(s)\ ds.$$

Suppose ΔB has only one nonzero entry Δb_{ij}, that is

$$\Delta B = \begin{bmatrix} & 0 & \\ 0 & \Delta b_{ij} & 0 \\ & 0 & \end{bmatrix}$$

and $\Delta A = 0$. Then using the foregoing procedure it follows that

$$\sum_{i,j} b_{ij} \frac{\partial X(t)}{\partial b_{ij}} = \int_o^t [\exp(A)(t-s)]\ BU(s)\ ds \quad .$$

Adding the two sensitivity function sums results in,

$$\sum_{i,j} \left(a_{ij} \frac{\partial X(t)}{\partial a_{ij}} + b_{ij} \frac{\partial X(t)}{\partial b_{ij}} \right) = \int_o^t [\exp(A)(t-s)][AX(s) + BU(s)]ds$$

which completes the proof.

An expression for the sum of second order sensitivity functions is determined in the following.

Theorem 3

If X(t) is the solution of the state equations of Theorem 2, then

$$\sum_{i,j} \sum_{m,n} \alpha_{ij}\ \alpha_{mn} \frac{\partial^2 X(t)}{\partial \alpha_{ij} \partial \alpha_{mn}} = 2A \int_o^t [\exp(A)(t-s)] \sum_{p,q} \alpha_{pq} \frac{\partial X(s)}{\partial \alpha_{pq}}\ ds$$

where $\alpha_{ij} = a_{ij}, b_{ij}$ for all i,j.

Proof: Suppose

$$F(t) = \int_o^t [\exp(A)(t-s)][AX(s) + BU(s)]ds \quad .$$

Then F(t) is the solution of

$$\dot{F}(t) = AF(t) + AX(t) + BU(t), \quad F(o) = 0.$$

Suppose $A \to A+\Delta A$, $B \to B+\Delta B$, $F(t) \to F(t)+\Delta F(t)$ where $F(o)+\Delta F(o) = 0$ and $X(t) \to X(t)+\Delta X(t)$. It follows that

$$\dot{F}(t) + \Delta\dot{F}(t) = (A+\Delta A)[F(t)+\Delta F(t)] + (A+\Delta A)[X(t)+\Delta X(t)] + (B+\Delta B)U(t)$$

which simplifies to

$$\Delta\dot{F}(t) = (A+\Delta A)\Delta F(t) + \Delta A[F(t)+X(t)] + (A+\Delta A)\Delta X + \Delta BU(t) \quad .$$

Therefore by Theorem 1,

$$\begin{aligned}\Delta F(t) = &\int_o^t [\exp(A+\Delta A)(t-s)][\Delta A(F(s)+X(s)) + \Delta BU(s)]ds \\ &+ \int_o^t [\exp(A+\Delta A)(t-s)] \Big[(A+\Delta A) \int_o^s [\exp(A+\Delta A)(s-\sigma)] \\ &[\Delta AX(\sigma) + \Delta BU(\sigma)]d\sigma\Big] ds \quad .\end{aligned}$$

Using the same method as that in the proof of Theorem 2, it follows that

$$\begin{aligned}\sum_{i,j} \left[a_{ij} \frac{\partial F(t)}{\partial a_{ij}} + b_{ij} \frac{\partial F(t)}{\partial b_{ij}}\right] = &\int_o^t [\exp(A)(t-s)][A(F(s) \\ &+ X(s)) + BU(s)]ds \\ &+ A \int_o^t [\exp(A)(t-s)] \ F(s) \ ds \\ = \ &2A\int_o^t [\exp(A)(t-s)]F(s)ds + F(t) \quad .\end{aligned}$$

Since the left hand side of the foregoing equation can be written as

$$\sum_{i,j} \sum_{m,n} \alpha_{ij} \alpha_{mn} \frac{\partial^2 X(t)}{\partial\alpha_{ij}\partial\alpha_{mn}} + \sum_{i,j} \alpha_{ij} \frac{\partial X(t)}{\partial\alpha_{ij}}$$

where $\alpha_{ij} = a_{ij}, b_{ij}$ for all i,j, the conclusion follows.

III. INVARIANTS

The result of Theorem 2 is specialized to homogeneous systems in the following theorem.

Theorem 4

Consider the state equations

$$\dot{X}(t) = AX(t), \qquad x(o) = X_o \neq 0$$

where $A = [a_{ij}]$. The sum of the (unnormalized) sensitivity functions always satisfy,

$$1. \quad \sum_{i,j} a_{ij} \frac{\partial X(t)}{\partial a_{ij}} = t \ \dot{X}(t)$$

for all i,j, and

$$2. \quad \sum_{i,j} S_{a_{ij}}^{X_k(t)} = S_t^{X_k(t)} \quad , \qquad k = 1, 2, \ldots, n$$

where $X^T(t) = [X_1(t), X_2(t), \ldots X_n(t)]$ and in general the normalized sensitivity function $S_v^u = \frac{v}{u}\frac{\partial u}{\partial v}$.

Proof: From Theorem 2 with U(t) = 0,

$$\sum_{i,j} a_{ij} \frac{\partial X(t)}{\partial a_{ij}} = \int_o^t [\exp(A)(t-s)] \ AX(s)ds \quad .$$

Since the solution $X(t) = [\exp(A)(t)]X_o$, it follows that

$$\sum_{i,j} a_{ij} \frac{\partial X(t)}{\partial a_{ij}} = \int_o^t [\exp(A)(t-s)]A[\exp(A)(s)]X_o \ ds \quad .$$

Since [exp(A)(t-s)] and A commute and also (A)(t-s) and (A)(s) commute,

$$\begin{aligned}\sum_{i,j} a_{ij} \frac{\partial X(t)}{\partial a_{ij}} &= A \int_o^t [\exp[(A)(t-s) + (A)(s)]]ds \ X_o \\ &= A \int_o^t [\exp(A)(t)]ds \ X_o\end{aligned}$$

$$= t\,A[\exp(A)(t)]\,X_o$$
$$= t\,\dot{X}(t) \quad .$$

This establishes 1. of the conclusion. The second part of the conclusion follows from the definition of the normalized sensitivity function given.

Suppose two different homogeneous linear systems have the same state vector X(t). Then by 2. of Theorem 4, the sum of the sensitivity functions $\sum_{i,j} S_{a_{ij}}^{X_k(t)}$ for both systems is the same and dependent only on t.

If $\sum_{i,j} a_{ij}\frac{\partial X(t)}{\partial a_{ij}} = 0$, $t \neq 0$, for some homogenous linear system, then by 1. of Theorem 4, $\dot{X}(t) = 0$. Therefore A is singular and not all the parameter in A which characterize the system are linearly independent.

Many frequency domain sensitivity invariants are known [1]. A sensitivity invariant is established in the following theorem by considering more detailed variations in the system matrices.

Theorem 5

Consider the state equations of a linear system,

$$\dot{X}(t) = M^{-1}N\,X(t) + M^{-1}P\,U(t), \qquad X(o) = X_o$$

where $M = [m_{ij}]$, $N = [n_{ij}]$ and $P = [p_{ij}]$. The sum of the sensitivity functions satisfy,

$$\sum_{i,j} \alpha_{ij}\frac{\partial X(t)}{\partial \alpha_{ij}} = 0$$

where $\alpha_{ij} = m_{ij}, n_{ij}$ and p_{ij} for all i,j .

Proof: By Theorem 1

$$\Delta X(t) = \int_o^t [\exp(M^{-1}N + \Delta(M^{-1}N))(t-s)][\Delta(M^{-1}N)X(s) + \Delta(M^{-1}P)\,U(s)]ds \quad .$$

The variations of the inverse of a matrix and the product of two matrices are expressed as follows,

$$\begin{aligned} \Delta M^{-1} &= -M^{-1}\cdot\Delta M\cdot M^{-1} \\ \Delta(M^{-1}N) &= \Delta(M^{-1})\cdot N+M^{-1}\cdot\Delta N = -M^{-1}\cdot\Delta M\cdot M^{-1}\cdot N+M^{-1}\cdot\Delta N \\ \Delta(M^{-1}P) &= \Delta(M^{-1})\cdot P+M^{-1}\cdot\Delta P = -M^{-1}\cdot\Delta M\cdot M^{-1}\cdot P+M^{-1}\cdot\Delta P \end{aligned}$$

A procedure similar to that used in the proof of Theorem 2 is now used. Suppose

$$\Delta N = \begin{bmatrix} & 0 & \\ 0 & \Delta n_{ij} & 0 \\ & 0 & \end{bmatrix}$$

and $\Delta P = 0$, $\Delta M = 0$. Then

$$\frac{\Delta X(t)}{\Delta n_{ij}} = \int_o^t [\exp(M^{-1}N+M^{-1}\Delta N)(t-s)]M^{-1} \begin{bmatrix} & 0 & \\ 0 & 1_{ij} & 0 \\ & 0 & \end{bmatrix} X(s)\,ds$$

and taking the limit as $\Delta n_{ij} \to 0$ and adding all such variations of N results in

$$\sum_{i,j} n_{ij}\frac{\partial X(t)}{\partial n_{ij}} = \int_o^t [\exp(M^{-1}N)(t-s)]M^{-1}N\,X(s)\,ds \quad .$$

For

$$\Delta P = \begin{bmatrix} & 0 & \\ 0 & \Delta p_{ij} & 0 \\ & 0 & \end{bmatrix}$$

and $\Delta M = 0$, $\Delta N = 0$, the same procedure results in,

$$\sum_{i,j} p_{ij} \frac{\partial X(t)}{\partial p_{ij}} = \int_o^t [\exp(M^{-1}N)(t-s)]M^{-1}P\ U(s)\ ds \quad .$$

For

$$\Delta M = \begin{bmatrix} & 0 & \\ 0 & \Delta m_{ij} & 0 \\ & 0 & \end{bmatrix}$$

and $\Delta N = 0$, $\Delta P = 0$, the same procedure results in,

$$\sum_{i,j} m_{ij} \frac{\partial X(t)}{\partial m_{ij}} = \int_o^t [\exp(M^{-1}N)(t-s)][-M^{-1}NX(s) - M^{-1}PU(s)]\ ds \quad .$$

Adding the three summations results in the conclusion.

That the conclusion of Theorem 5 holds in the frequency domain is easily established. Calculating the Laplace Transform of both sides of the conclusion gives,

$$\sum_{i,j} \alpha_{ij} \frac{\partial X(s)}{\partial \alpha_{ij}} = 0 \quad .$$

Also straightforward differentiation of the conclusion in Theorem 5 results in the following.

$$\sum_{i,j} \sum_{m,n} \alpha_{ij}\alpha_{mn} \frac{\partial^2 X}{\partial \alpha_{ij} \partial \alpha_{mn}} = 0$$

which hold in the time and frequency domain.

IV. COMMON FACTORS AND BOUNDS

A relation between sensitivity and observability and controllability of a system is established below.

Theorem 6

Consider the system

$$\dot{X}(t) = AX(t) + BU(t)$$
$$Y(t) = CX(t)$$

where its transfer function is $T(s) = \frac{P(s)}{Q(s)}$. If $S_{pi}^{T(s)} = 0$, where p_i is any simple zero of $Q(s)$ such that $p_i \neq 0$ and $s \neq \infty$, then

1. $s-p_i$ is a common factor in $T(s)$, and
2. the system is either unobservable or uncontrollable or both.

Proof: Suppose $s-p_i$ is not a common factor of $T(s)$. Then

$$S_{p_i}^{T(s)} = \frac{p_i}{T(s)} \frac{\partial}{\partial p_i} T(s) = \frac{-p_i \frac{\partial}{\partial p_i} Q}{Q(s)}$$

For $Q(s) = (s-p_i)Q_1(s)$,

$$S_{p_i}^{T(s)} = \frac{p_i}{s-p_i} \neq 0 \quad .$$

Since this is a contradiction, $s-p_i$ is a common factor of $T(s)$. The second part of the conclusion follows [5] from part 1.

An upper bound on the magnitude of the sensitivity sum of Theorem 2 will be determined next. In Theorem 2 it is determined that

$$F(t) = \sum_{i,j} \alpha_{ij} \frac{\partial X(t)}{\partial \alpha_{ij}} = \int_{o}^{t} [\exp(A)(t-s)] \frac{d}{ds} X(s)\, ds \quad .$$

Therefore

$$\frac{d}{dt} F(t) = \dot{X}(t) + A\, F(t) \quad .$$

Suppose $\frac{d}{dt} F(t) = 0$ for $t = t_i$, $i = 1, 2, \ldots, n$ and A^{-1} exists. Then

$$F(t_i) = -A^{-1} \dot{X}(t_i)$$

and if $\max_i |F(t_i)| = |A^{-1} \dot{X}(t_{io})|$ then

$$|F(t)| = \left| \sum_{i,j} \alpha_{ij} \frac{\partial X(t)}{\partial \alpha_{ij}} \right| \leq |A^{-1} \dot{X}(t_{io})| \text{ for all } t.$$

V. REFERENCES

1. M.N.S. Swamy, C. Bhushan and K. Thulsiraman,"Sensitivity invariance for linear time-invariant networks," IEEE Trans. CT-20, pp. 21-24, 1973.

2. I.M. Horowitz and U. Shaked, "Superiority of transfer function over state-variable methods in linear time-invariant feedback system design," IEEE Trans. AC-20, pp. 84-97, 1975.

3. B.S. Morgan, Jr., "Sensitivity analysis and synthesis of multivariable systems," IEEE Trans. AC-11, pp. 506-512, 1966.

4. M. Lal, J.S. Sohal and H. Singh, "Sensitivity invariance using state variables," Proc. 12th Allerton Conference on Circuit and System Theory, Allerton, Illinois, pp. 967-974, 1974.

5. B.C. Kuo, Discrete Data Control Systems, Prentice-Hall, New Jersey, 1970.

SEQUENCE GENERATION BY AUTONOMOUS DISCRETE-TIME SYSTEMS: FURTHER RESULTS

MARTIN E. KALISKI
Department of Electrical Engineering
Northeastern University
Boston, Massachusetts 02115

QUENTIN L. KLEIN
Signal Processing Systems, Inc.
Waltham, Massachusetts 02154

ABSTRACT

This paper considers the following problems: (i) generating with autonomous discrete-time systems of one-dimension classes of sequences in the interval [0,1] exhibiting specific topological characteristics, and (ii) developing measures of sequence complexity and exploring the relationships between such measures and sequence generability. Partial results are obtained in each area of investigation.

I. INTRODUCTION

The general problem under consideration may be stated as follows: given a bounded sequence of real numbers, $\hat{r} = r_0\ r_1\ r_2 \ldots$, and given real constants $L > 0$, $B > 0$, find a natural number $n > 0$, a function $\underline{G}:R^n \to R^n$ (R^n is n-dimensional real-vector space, in the usual topology), and an $\underline{x}_0$ in R^n such that the n-dimensional discrete-time system with function block $\underline{G}$ and initial state $\underline{x}_0$ autonomously generates $\hat{r}$, (Figure 1) i.e. such that

(i) For $i = 0,1,2,\ldots$

$$r_i = \pi_1 \underline{x}_i$$

$$\underline{x}_{i+1} = \underline{G}(\underline{x}_i)$$

(with $\pi_1:R^n \to R$ the projection map $\pi_1(a_1,\ldots,a_n) = a_1$).

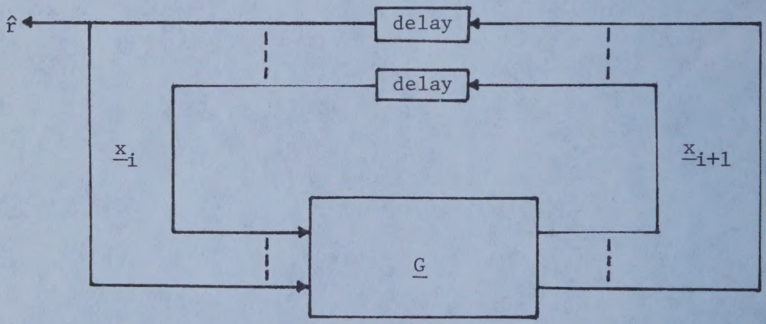


Figure 1: Generating a Sequence $\hat{r}$

Furthermore require that $\underline{G}$ obey the following two idealized physical constraints:

(ii) (L-Lipschitzness) For all $\underline{x} = (x_1,\ldots,x_n)$, $\underline{y} = (y_1,\ldots,y_n)$ in R^n and for $j = 1,\ldots,n$,

$$|\pi_j\underline{G}(\underline{x})-\pi_j\underline{G}(\underline{y})| \le L(|x_1-y_1| + |x_2-y_2| + \dots + |x_n-y_n|).$$

(Here $\pi_j : R^n \to R$ is the projection map $\pi_j(a_1,\dots,a_j,\dots,a_n)=a_j$.)

(iii) (B-boundedness) For all $\underline{x}$ in R^n and for $j = 1,\dots,n$, $|\pi_j\underline{G}(\underline{x})| \le B$.

The work described in this paper extends the previous work of one of the authors [KALISKI, 1974] on this problem.

In this earlier study the following issues were addressed:

(A) Necessary and sufficient conditions for generability of sequences in one-dimension ($n = 1$) with given L and B.

(B) An approach to generability in higher dimensions by means of a process called sequence augmentation.

(C) Approaches towards generating certain special classes of sequences.

(D) The development of several non-generability results: results showing that given sequences cannot be generated with given n, L, and B.

This paper focuses on two additional aspects of the generability question that were among those left unaddressed in the earlier work:

(i) Is there a relationship between the topology of a sequence's point set and the sequence's generability in one-dimension?

(ii) What kind of numerical measures of a sequence's complexity can be developed that relate to generability?

Partial answers to these questions will be developed in the next two sections. In considering the latter question, the concept of sequence augmentation, referred to in (B), above, will be re-introduced.

II. TOPOLOGY AND SEQUENCE GENERABILITY IN ONE-DIMENSION: MONOTONE SEQUENCES AND DENSE SEQUENCES IN [0,1]

In an effort to relate the topology of the point set $S(\hat{r}) = \{r_0, r_1, r_2, \dots\}$ of a sequence $\hat{r}$ to its generability in one-dimension, we consider two classes of sequences with "extreme" topologies: monotone sequences, whose point sets have single limit points, and dense sequences, whose point sets are dense in [0,1].

Note that a sequence $\hat{r} = r_0 r_1 r_2 \dots$ has a one-dimensional generation with given $L, B > 0$ if and only if there is a $\underline{G}:R \to R$, $\underline{G}$ L-Lipschitz and B-bounded, for which $r_{i+1} = \underline{G}(r_i)$ for $i = 0,1,2,\dots$, and that this is equivalent to the condition: for all $i,j \ge 0$

(i) $|r_{i+1} - r_{j+1}| \le L|r_i - r_j|$

(ii) $|r_{i+1}| \le B$

[KALISKI, 1974]

We can dismiss immediately any hope of a relationship between topology and generability for the case of monotone sequences. The sequence $\hat{r} = 1 \; 1/2 \; 1/4 \; 1/8 \dots 1/2^n \dots$ clearly obeys (i) and (ii) above, with $B = 1/2$, $L = 1/2$ and is thus generable in one-dimension. However, as will be seen in section III, $\hat{r} = 1/2+1/4 \quad 1/2 \quad 1/4+1/16 \quad 1/4 \dots 1/2^j \quad 1/2^{j+1}+1/2^{2j+2} \quad 1/2^{j+1} \dots$ cannot be generated in one-dimension with any finite $L > 0$. We therefore devote the remainder of this section to dense sequences.

To say that $\hat{r}$ is dense means that every point α in [0,1] is either equal to r_j, for some $j \ge 0$, or has the property that for any $\varepsilon > 0$ there is a term r_j in $\hat{r}$ for which $|\alpha - r_j| \leq \varepsilon$. Typical dense sequences are (i) a sequence consisting of all the rational numbers in (0,1) in some fixed enumeration, and (ii) a sequence consisting of all numbers of the form $k/3^m$, $m = 0,1,2,\ldots$, $0 \leq k \leq 3^m-1$, in some fixed enumeration. Unfortunately, as in the case of monotone sequences, a direct relationship between the topology of $S(\hat{r})$ and the generability of $\hat{r}$ does not seem likely for the case of dense sequences.

In what follows we present two results. We first show that a particular dense sequence can be generated in one-dimension with L = 2 and B = 2, via a particularly simple function $\underline{G}$, and we then show that another dense sequence, with a particularly simple description (i.e., with a simple rule for the $n^{\underline{th}}$ term) cannot be generated in one-dimension with any finite $L > 0$, and any $B > 0$.

Theorem 1: Let $\underline{G}:R \to R$ be the map

$$\underline{G}(x) = \begin{cases} 0 & \text{if } x < 0 \\ 2x & \text{if } 0 \leq x < 1/2 \\ 2-2x & \text{if } 1/2 \leq x \leq 1 \\ 0 & \text{if } x > 1 \end{cases}$$

Then $\underline{G}$ is L-Lipschitz with L = 2, B-bounded with B = 2, and the sequence of iterates $\hat{r} = r_0\ G(r_0)\ G(G(r_0))\ldots$ is dense in [0,1], with r_0 chosen to be

$$r_0 = (.00010000001010011000000001001000110100010101100111\ldots)_2$$

(r_0 is a binary fraction formed as follows. Write down all finite binary strings in normal enumerative order, preceding each by a 0: $0\underline{0}$ $0\underline{1}$ $0\underline{00}$ $0\underline{01}$ $0\underline{10}$ $0\underline{11}$ $0\underline{000}$ $0\underline{001}\ldots$)

In terms of Picard iteration [Ortega and Rheinboldt, 1970] this Theorem says that a relatively simple function may have a dense set of iterates (Figure 2).

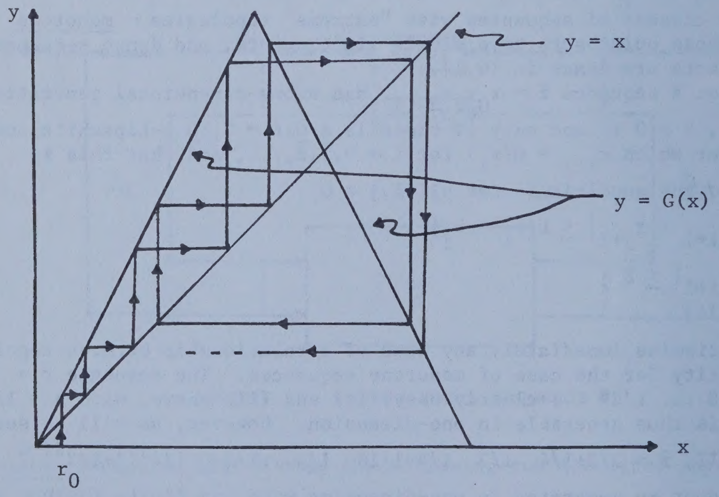


Figure 2: The Iterates of r_0 Under $\underline{G}$

Proof: We state the following Lemma without proof. It tells us how to calculate the iterates of $\underline{G}$, $\underline{G}^j(x) = \underline{G}(\underline{G}(\underline{G}(\ldots(x))))$ (j-times), $j = 2,3,4,\ldots$ by means of reflections and modular reductions. For the purposes of the Lemma, $\underline{G}^1(x)$ should be taken to mean $\underline{G}(x)$.

Lemma: For any $j \geq 1$, x in $[0,1]$, we have:

$$\underline{G}^j(x) = R(2^j x \bmod 2)$$

where $R\colon[0,2] \to [0,1]$ is given by

$$R(y) = \begin{cases} y \text{ if } 0 \leq y < 1 \\ 2-y \text{ if } 1 \leq y \leq 2 \end{cases}$$

(reflection)

and, where for any $\alpha \geq 0$, $\alpha \bmod 2$ is defined as $\alpha - 2\ell$, where ℓ is the largest integer such that $2\ell \leq \alpha$ (modular reduction).

Using this Lemma we can show that $\hat{r}$ is dense as follows:

Consider the terms of $\hat{r}$, $r_0, r_1, r_2, \ldots, r_j = \underline{G}^j(r_0)$. Let us begin to write down some of them explicitly.

$$r_1 = G(r_0) = R(2r_0 \bmod 2) = .0010000010100110000001\ldots$$

$$r_2 = G^2(r_0) = R(4r_0 \bmod 2) = .0100001010\ldots$$

$$r_3 = G^3(r_0) = R(2^3 r_0 \bmod 2) = .1000001010\ldots$$

$$r_{15} = G^{15}(r_0) = R(2^{15} r_0 \bmod 2)$$

$$= R(100000101001.100000001\ldots \bmod 2)$$

$$= R(1.100000001\ldots)$$

$$= .011111110\ldots$$

It should be clear that to determine r_j, $j \geq 1$, we move the binary point in r_0 j places to the right and remove all but the single bit to the left of the binary point (reduction mod 2). Then, if the remaining bit to the left of the point is a 0, do nothing more; if it is a 1, complement the binary number, bit by bit (applying the R function). The result is r_j. By virtue of our choice of r_0 it should be clear that any desired finite length binary string of length k appears as the initial k bits of some r_j. It follows that $\hat{r}$ is dense.

This completes the proof of Theorem 1 as it is obvious that $\underline{G}$ is L-Lipschitz and B-bounded with $B = L = 2$. Q.E.D.

Remark: It is possible to show (it will not be done here) that all denumerable dense sets in (0,1) are, in some sense, "equivalent" from the point of view of Lipschitz mappings: specifically, if D_1 and D_2 are two denumerable dense subsets of (0,1) and $L > 1$ is arbitrary, then there is a 1-1 onto map $g\colon R \to R$ for which

(i) g and g^{-1} are L-Lipschitz, B-bounded maps, with $B = 1$.

(ii) $g(D_1) = D_2$

and (iii) $g(0) = 0$ and $g(1) = 1$.

It then follows, using Theorem 1, that there is some enumeration of the rational numbers in (0,1) that can be generated with any desired $L > 2$, with $B = 2$, in one-dimension:

Corollary: Let $L_0 > 2$ be arbitrary. Then there is an enumeration of the rationals in (0,1) that can be generated in one-dimension with $L = L_0$, $B = 2$.

Proof: Let $S(\hat{r})$ be the point set associated with the sequence

$\hat{r} = r_0 r_1 r_2 \ldots$ of Theorem 1. Let Q be the rational numbers in (0,1). By our remark above there is a map $g: R \to R$ obeying (i) - (iii) above for $D_1 = Q$, $D_2 = S(\hat{r})$ (recall that $S(\hat{r})$ is dense, by Theorem 1) and with $L = \sqrt{L_0/2}$. Consider the sequence $r^* = r_0^* \, r_1^* \, r_2^* \ldots$ defined by $r_i^* = \underline{\tilde{G}}^i(r_0^*)$ with $\underline{\tilde{G}}(x) = g^{-1}(\underline{G}(g(x))$, for all x in R, where $\underline{G}$ is the function of Theorem 1, and with $r_0^* = g^{-1}(r_0)$. Then r^* is the sequence $r^* = g^{-1}(r_0) \, g^{-1}(r_1) \, g^{-1}(r_2) \ldots$. By our observations, r^* is thus an enumeration of Q. It is generable in one-dimension with function $\underline{\tilde{G}}$ which has Lipschitz constant $\leq \sqrt{L_0/2} \cdot 2 \cdot \sqrt{L_0/2} = L_0$ and has bound $B = 2$. Q.E.D.

Theorem 2: There is an enumeration of the rationals in (0,1), in which no term is repeated, which is not generable in one-dimension with any finite $L > 0$.

Proof: (Note that any enumeration of the rationals in (0,1) in which some rational occurs more than once is non-generable in one-dimension. This follows from the cited earlier work [KALISKI, 1974].)

Consider $\hat{r} = 1/2 \;\; 2/3 \;\; 1/3 \;\; 1/4 \;\; 3/4 \;\; 4/5 \;\; 3/5 \;\; 2/5 \;\; 1/5 \;\; 1/6 \;\; 5/6 \ldots$ (all terms distinct). If m is even and > 2 then $r_i = 1/m$ has successor $r_{i+1} > 2/m$, and $r_j = 1/m{+}1$ has successor $r_{j+1} = 1/m{+}2$,

$$\text{whence} \quad \frac{|r_{i+1} - r_{j+1}|}{|r_i - r_j|} \geq (m + 4) \, \frac{(m + 1)}{(m + 2)} \; .$$

As m is arbitrary there is no finite L for which $|r_{i+1} - r_{j+1}| \leq L|r_i - r_j|$ for all $i,j \geq 0$.

Thus $\hat{r}$ does not have a one-dimensional generation. Q.E.D.

Note that to form $\hat{r}$ one must simply count up and then count down repeatedly with successively higher fraction denominators, avoiding duplication.

III. MEASURES OF SEQUENCE COMPLEXITY:
MONOTONICITY, SEQUENTIAL VARIATION AND ADJACENCY VARIATION

By any reasonable sense of the word "complexity", dense sequences are ill-behaved. They exhibit no properties like monotonicity or convergence, and possess infinitely many subsequential limits. In this section we explore several possible ways of defining "good behavior" in a sequence and show that, at least for two of our definitions, "good behavior" implies a great deal about sequence generability. The third of our definitions, adjacency variation, is of an experimental nature, and will be briefly explored below.

It is useful at this point to re-introduce the concept of sequence augmentation (KALISKI, 1974). If $\hat{r} = r_0 r_1 r_2 \ldots$ cannot be generated in one-dimension with any finite $L > 0$, it may, nonetheless, be possible to generate $\hat{r}$ in two or more dimensions by augmenting other sequences to $\hat{r}$. These sequences can serve to take points in $\hat{r}$ that are close together with "far apart" successors and "pull them apart" in the other dimensions. We refer the reader to the cited work for a further discussion of this. Suffice it to say that in a two-dimensional context, we have: (KALISKI, 1974) $\hat{r} = r_0 r_1 r_2 \ldots$ can be generated in two-dimensions with Lipschitz constant L, bound B if and only if there exists $\hat{s} = s_0 s_1 s_2 \ldots$ for which:

(i) $|r_{i+1} - r_{j+1}| \le L(|r_i - r_j| + |s_i - s_j|)$

$|s_{i+1} - s_{j+1}| \le L(|r_i - r_j| + |s_i - s_j|)$

(ii) $|r_{i+1}|, |s_{i+1}| \le B$ for all $i,j \ge 0$.

Monotone Sequences

The simplest form of "good behavior" we will consider is monotonicity.

Definition 1: $\hat{r} = r_0 r_1 r_2 \ldots$ is monotone if for all $i \ge 0$, $r_{i+1} \le r_i$. (Without loss of generality it suffices to study monotone decreasing sequences.)

There are two facts of particular interest about monotone sequences.

Theorem 3: There is a monotone sequence that cannot be generated in one-dimension with any finite $L > 0$. However all monotone bounded sequences can be generated in two dimensions with any desired $L > 1$.

Proof: For the first part of the Theorem consider $1/2+1/4$ $1/2$ $1/4+1/16$ $1/4$ $1/8+1/64$ $1/8$ ... $1/2^j$ $1/2^{j+1}+1/2^{2j+2}$ $1/2^{j+1}$ .... The terms $1/2^{j+1}+1/2^{2j+2}$ and $1/2^{j+1}$, for any $j > 0$ have successors, $1/2^{j+1}$ and $1/2^{j+2}+1/2^{2j+4}$, respectively. If $\hat{r}$ is to be generable in one-dimension with Lipschitz constant $L > 0$ it follows, that for all $j > 0$,

$$L \ge \frac{\frac{1}{2^{j+1}} - \frac{1}{2^{j+2}} - \frac{1}{2^{2j+4}}}{\frac{1}{2^{2j+2}}} = 2^{j+1} - 2^j - \frac{1}{4}$$

Clearly then, for all $j > 0$, $L \ge 2^j - 1/4$ and L cannot be finite.

As for the second part, let $L > 1$ be given. Assume $|r_i| \le B$ for all $i \ge 0$. Define for $j \ge 0$

$$q_j = \sup_{k \ge 0} \frac{r_{j+k} - r_{j+k+1}}{L^k}$$

Observe that q_j is well-defined, is positive, and that $\sum_{j=0}^{\infty} q_j$ is finite: for given M, N arbitrary positive integers, $\sum_{j=0}^{M} \sup_{N \ge k \ge 0} \frac{r_{j+k} - r_{j+k+1}}{L^k} \le$

$\sum_{j=0}^{M} \sum_{k=0}^{N} \frac{r_{j+k} - r_{j+k+1}}{L^k} = \sum_{k=0}^{N} \sum_{j=0}^{M} \frac{r_{j+k} - r_{j+k+1}}{L^k} = \sum_{k=0}^{N} \frac{1}{L^k}(r_k - r_{M+k+1}) \le$

$2B \sum_{k=0}^{N} \frac{1}{L^k} \le \frac{2BL}{L-1}$. As M,N are arbitrary it follows that $\sum_{j=0}^{\infty} q_j$ is finite,

$= \beta$, for some $\beta > 0$.

We will choose the augmenting sequence $\hat{s} = s_0 s_1 s_2 \ldots$ to be monotone itself according to

$s_0 = 0$

$$s_i = \left(-\sum_{\ell=0}^{i-1} q_\ell\right) - (r_i - r_0)$$

(Note that for $i \geq 0$, $|s_i| \leq \beta + 2B$. Thus $\hat{s}$ is bounded. Also note that as for any $m \geq 0$, $s_m - s_{m+1} = q_m - (r_m - r_{m+1})$ and $q_m \geq r_m - r_{m-1}$, it follows that $s_m - s_{m+1} \geq 0$. Whence $\hat{s}$ is indeed monotone.) Observe therefore that for $i,j \geq 0$, and $i > j$, we have

$|r_{i+1} - r_{j+1}| = r_{j+1} - r_{i+1} = (r_{j+1} - r_{j+2}) +\ldots+ (r_i - r_{i+1}) \leq q_{j+1} +\ldots+ q_i$, and $|s_{i+1} - s_{j+1}| = s_{j+1} - s_{i+1} = (s_{j+1} - s_{j+2}) +\ldots+ (s_i - s_{i+1}) = q_{j+1} + (-r_{j+1} + r_{j+2}) +\ldots+ q_i + (-r_i + r_{i+1}) = q_{j+1} +\ldots+ q_i + (-r_{j+1} + r_{i+1}) \leq q_{j+1} +\ldots+ q_i$, as well.

Also note that $|r_i - r_j| + |s_i - s_j| = r_j - r_i + s_j - s_i = (r_j - r_{j+1}) + (s_j - s_{j+1}) +\ldots+ (r_{i-1} - r_i) + (s_{i-1} - s_i) = q_j +\ldots+ q_{i-1}$.

Therefore, as for all $m, q_{m+1} \leq Lq_m$, we have for all $i,j \geq 0$, $i < j$

$$|r_{i+1} - r_{j+1}| \leq L(|r_i - r_j| + |s_i - s_j|)$$

$$|s_{i+1} - s_{j+1}| \leq L(|r_i - r_j| + |s_i - s_j|)$$

as desired. Q.E.D.

Monotone sequences are a very constrained class of sequences and although Theorem 3 is of great interest, we seek a more general measure of "good behavior".

Sequential Variation of Sequences

Definition 2: The sequential variation (s.v.) of a sequence $\hat{r}$ is the sum

$$\sum_{i=0}^{\infty} |r_{i+1} - r_i|$$

If this sum exists we will say that $\hat{r}$ has finite s.v. Observe that any monotone bounded sequence has finite s.v. and that any sequence with finite s.v. is convergent. A typical sequence with finite s.v. is $\hat{r}$ = 1 1-1/2 1-1/2+1/4.... We claim that Theorem 3 has an analogue for the finite s.v. case, with the value on L changed, however.

Theorem 4: All sequences with finite s.v. can be generated in two-dimensions with any desired $L > 2$.

Proof: As the proof is analogous to that of Theorem 3, we simply sketch the essential differences.

(i) q_j will be defined as $q_j = \sup_{k \geq 0} \dfrac{|r_{j+k} - r_{j+k+1}|}{L^k}$

(ii) $\hat{s}$ will be defined by $s_0 = 0$ $s_i = \left(\sum_{\ell=0}^{i-1} q_\ell\right) - (r_i - r_0)$

(iii) In the inequality for $|s_{i+1} - s_{j+1}|$ we reach the line

$|s_{i+1} - s_{j+1}| = (q_{j+1} +\ldots+ q_i) + (r_{j+1} - r_{i+1})$ and can at best continue this to $\leq 2(q_{j+1} +\ldots+ q_i)$.

Except for some differences in signs the essential manipulations are unchanged. By virtue of the "2" introduced above, we can only guarantee that $L > 2$ works. Q.E.D.

Adjacency Variation of Sequences

We conclude this section by introducing a more encompassing measure of sequence complexity - adjacency variation. It is of an experimental nature and is an attempt to measure the spatial variation of a sequence.

Recall that if $f:D \subseteq R \to R$ is an arbitrary function, D an arbitrary subset of R, the variation of f is defined by

$$\text{variation } (f) = V(f) = \sup_{D_0 \subseteq D} \sum_{i=1}^{\#D_0 - 1} |f(d_{i+1}) - f(d_i)|$$

where the sup is taken over all finite subsets D_0 of D, where $\#D_0$ is the number of points in D_0 and where these points are ordered $d_1 < d_2 < \ldots < d_{\#D_0}$.

Observe that if f is L-Lipschitz on D and if D is a bounded subset of R, $D \subseteq [-B,B]$ then

$$V(f) \le \sup_{D_0 \subseteq D} L \sum_{i=1}^{\#D_0 - 1} |d_{i+1} - d_i| \le L \sup_{D_0 \subseteq D} 2B \le 2LB$$

Thus any L-Lipschitz function over a bounded domain has finite variation.

Definition 3: Let $\hat{r} = r_0 r_1 r_2 \ldots$ be a given bounded sequence, all of whose terms are distinct. Let $S(\hat{r})$ be the point set associated with $\hat{r}$. Let $f:S(\hat{r}) \to R$ be defined by the algorithm: if α is an element of $S(\hat{r})$ and $\alpha = r_i$ for some i, then $f(\alpha) = r_{i+1}$. We define the adjacency variation of $\hat{r}$ to be the variation of f, $V(f)$. If $V(f)$ is finite, $\hat{r}$ will be said to have finite a.v.

Lemma: If $\hat{r}$ has finite s.v. it has finite a.v.

Proof: Assume $|r_i| \le B$ for $i \ge 0$. Consider for any $D_0 \subseteq S(\hat{r})$,

$$\sum_{i=1}^{\#D_0 - 1} |f(d_{i+1}) - f(d_i)|.$$ By the triangle inequality this is,

$$\le \sum_{i=1}^{\#D_0 - 1} |f(d_{i+1}) - d_{i+1}| + \sum_{i=1}^{\#D_0 - 1} |d_{i+1} - d_i| + \sum_{i=1}^{\#D_0 - 1} |f(d_i) - d_i|.$$

The first and last sum, by the definition of f, are bounded above by the s.v. of $\hat{r}$, for all choices of D_0. The middle sum is bounded by $2B$. Therefore,

$$\text{a.v.}(\hat{r}) \le \text{s.v.}(\hat{r}) + 2B + \text{s.v.}(\hat{r}).$$

The result follows. Q.E.D.

Clearly, by Definition 3, if $\hat{r}$ has a one-dimensional generation, it has finite a.v., since f is L-Lipschitz on $S(\hat{r})$, for some $L > 0$.

Remark: By virtue of section II it follows that there are dense sequences with finite a.v. It is our feeling that a.v. represents a useful characterization of sequence complexity, bearing a possible relationship to generability of sequences in higher dimensions.

IV. CONCLUSIONS

This paper has considered several aspects of the problem of sequence generation, involving topological and variational notions. Current activities of the authors are directed towards developing a complete characterization of generability in one and two dimensions.

REFERENCES

(1) Kaliski, Martin E. (1974) Autonomous Sequence Generation, Information and Control, 26, 3.

(2) Ortega, J.M. and Rheinboldt, W.C. (1970) Iterative Solution of Non-Linear Equations in Several Variables, Academic Press.

EXPLICIT CHARACTERIZATION OF THE STABILIZABILITY REGION OF LINEAR TIME-INVARIANT SYSTEMS

FRANCISCO D. GALIANA
University of Michigan
Ann Arbor, Mich.

ABSTRACT

For $\dot{x} = A(g)x$; $A(g)=A_o + \sum_{k=1}^{m} g_k A_k$; A_k, $k = 0, 1, \ldots, m$ given, we seek the values of g for which the system will be stable or achieve a certain damping. Those g-values for which an $M > 0$ satisfies $A^T(g)M + MA(g) \leqslant 0$ stabilize $A(g)$. Moreover, this set is convex. A technique has been developed for determining the extreme values of this set using linear programming.

INTRODUCTION

This paper describes a technique for explicitly characterizing convex subsets of the stabilizability region of multivariable linear time invariant dynamic systems. The parameters used to stabilize the system are assumed to enter linearly in the system matrices, and as a consequence can represent many forms of output feedback control, both static and dynamic.

The problem considered here is then the explicit characterization of convex subsets of the set of parameters $g \in R^m$ with the property that all the eigenvalues of the matrix $A(g) \in R^{nxn}$ lie on or to the left of the real number -d. The dynamic system described by,

$$\dot{x} \;=\; A(g)\, x \tag{1}$$

where $x \in R^m$ is then said to have damping d. In this paper A(g) is assumed to be of the form

$$A(g) \;=\; A_o + \sum_{k=1}^{m} g_k A_k \tag{2}$$

where the parameters g_k, k=1, 2,... m are the elements of the vector g used to achieve a given damping, while A_k, k=o, 1, 2, ... m are n by n given constant matrices. The matrix A_o may represent the open-loop matrix while the rest are defined by the feedback structure. The stabilizability region, R_{gd}, is defined by all those elements $d \in R$ and $g \in R^m$ which generate a damping d in the system defined by equations (1) and (2).

The explicit characterization of convex subsets of R_{gd} should have theoretical and practical value. For example, the design of feedback controls subject to gain inequality constraints could be investigated based on such a characterization. Another possibility is the determination of the maximum damping achievable by a certain feedback control structure. The results obtained may also serve to analyze the damping capabilities of a feedback scheme prior to the use of more elaborate methods such as pole placement (2, 3), or they may serve to define initial gain values (regions) in optimization methods (4, 5). Often, however, because of its simplicity, damping alone may serve as a performance criterion (6).

The technique presented here is based on the Lyapunov conditions for the stability of linear systems (1). The results obtained so far indicate that the method yields sufficient but not necessary conditions for the existence of a convex set of gains g leading to a desired damping, d. If the method fails to generate such a set of gains it can imply that the system cannot achieve the desired damping (i.e. be d-stabilized) with the chosen feedback structure, or that the sufficient condition cannot be satisfied. Further research is needed to strengthen the sufficient conditions described in this paper. When the sufficient condition is satisfied, the result is a non-empty convex subset of the stabilizability region, R_{gd}, for the desired damping levels. This subset is characterized by the intersection of a number of planes in R^{m+1}, obtained by a succession of eigenvalue and linear programming steps.

PRELIMINARY BACKGROUND

The system defined by (1) and (2) has damping d for some g if and only if for all $Q \geqslant 0$, $Q \varepsilon R^{nxn}$, belonging to the range space defined by the left side of equation (3), there exists a matrix $M > 0$, $M \varepsilon R^{nxn}$, such that

$$(A(g) + d\ I)^T M + M(A(g) + d\ I) = -Q \leqslant 0 \qquad (3)$$

where I is the unit matrix in R^{nxn}. This is a consequence of the Lyapunov stability condition (1).

From the above it follows that if an $M > 0$ is found such that for some g, d and $Q \geqslant 0$, relation (3) can be satisfied, then the system will be d-stabilizable. Relation (3) with a given $M > 0$ is thus a sufficient condition for achieving a damping d through g.

If no g can be found which satisfies (3) for a desired d and the chosen $M > 0$, then this damping may or may not be attainable. Different values of M may still exist for which a g can be found which satisfies relation (3) for some $Q \geqslant 0$.

If the system is stabilizable for a given d and g then there exist an infinity of $M > 0$ which will satisfy (3). This follows since Q is positive semi-definite, but otherwise almost arbitrary. Thus for each $Q \geqslant 0$ (belonging to the range space of the left side of equation (3)) a matrix $M > 0$ is defined satisfying (3) for the given g and d.

The proposed method relies on the hypothesis that if the system is stabilizable then it would not be difficult to find an $M > 0$ sufficient to find a g which satisfies (3) for a given d. In this paper some procedures are suggested for obtaining a reasonable choice of $M > 0$.

SUFFICIENT CONDITIONS FOR D-STABILIZABILITY

Let $-d_o$ be the maximum real part of the eigenvalues of the open-loop matrix A_o. Thus d_o is the open-loop damping.

Let ε be a real parameter greater than zero. Then the matrix $A_o + d_o I$ has all its eigenvalues on the jw axis while the matrix $A_o + (d_o - \varepsilon) I$ has its eigenvalues to the left of the jw axis. A positive definite matrix M can then always be defined by the Lyapunov equation

$$(A_o + (d_o-\varepsilon)I)^T M + M(A_o + (d_o-\varepsilon)) = -I \qquad (4)$$

where I is the unit matrix in $R^{n\times n}$.

An $M > 0$ satisfying (4) will exist for any $\varepsilon > 0$. Substitution of (4) into the relation (3) yields

$$-I + 2(\varepsilon + d - d_o)M + \sum_{k=1}^{m} g_k(A_k^T M + M A_k) \leqslant 0 \qquad (5)$$

Relation (5) with the matrix $M > 0$ as defined by (4) for some $\varepsilon > 0$, constitutes the sufficient condition investigated in this study. Other, perhaps stronger, sufficient conditions (i.e. choices of M) should be further investigated. The motivation behind the particular choice of M defined in equation (4) was loosely as follows:

Suppose that a small g exists which pushes the eigenvalues of $A(g)+d_oI$ slightly to the left of the jw axis. Now for a small $\varepsilon>0$, the eigenvalues of $A_o+(d_o-\varepsilon)I$ also lie slightly to the left of the jw axis. The assumption is made then that since the two matrices $A(g)+d_oI$ and $A_o+(d_o-\varepsilon)I$ are "close" to each other, that the value of M defined by (4) can be made to satisfy (3) for some g, some $d > d_o$ and some $Q \geqslant 0$.

This assumption seems to be valid for a number of examples tried. In fact the requirement that ε and g be "small" is not essential.

If we now define g_o as

$$g_o = 2(\varepsilon + d - d_o) \qquad (6)$$

then the condition that the closed loop damping d be at least greater than d_o, the open loop damping, implies that

$$g_o > 2\varepsilon \qquad (7)$$

In order to have positive damping we must have

$$g_o > 2(\varepsilon - d_o) \qquad (8)$$

The above can now be summarized by the following theorem:

Theorem 1

Given the matrices A_i, i=0, 1, 2,... m, a parameter $\varepsilon > 0$ (possibly "small"), and the matrix $M > 0$ defined by (4), then if a $g_o \in R$ and $g \in R^m$ satisfy

$$-I + \sum_{k=0}^{m} g_k J_k \leqslant 0 \qquad (9)$$

where $J_o \triangleq M \qquad (10)$

and $J_k \triangleq A_k^T M + M A_k \qquad (11)$

the closed loop system defined by (1) and (2) has damping $g_o/2 + d_o - \varepsilon$. For increased closed loop damping we should also satisfy the condition

$$g_o > 2\varepsilon \qquad (12)$$

while for positive damping we should have

$$g_o > 2(\varepsilon - d_o) \qquad (13)$$

CHARACTERIZATION OF STABILIZABILITY REGION

This section describes a technique for explicitly characterizing the set of g_o and g satisfying the conditions of Theorem 1 summarized by equations (9), (10) and (11). This set is denoted by C_{gd}. After characterizing C_{gd} it is relatively simple to add the constraint given by (12), i.e. the requirement that damping should increase with the feedback gains, or (13), that it have positive damping.

Lemma 1

The set C_{gd} is convex.

Proof

This follows directly from the linear inequality (9) defining those g_k's belonging to C_{gd}.

Lemma 2

The set C_{gd} is non-empty.

Proof

Let $g_k=0$, for k=1, 2, ... m. Let g_o be sufficiently small, then

$$-I + g_o M \leqslant 0 \tag{14}$$

if

$$g_o \leqslant (\max_i \lambda_i(M))^{-1} \tag{15}$$

The notation $\lambda_i(M)$ signifies the i^{th} eigenvalue of the matrix M.

Note that the limiting value of g, $\left\{\max_i \lambda_i(M)\right\}^{-1}$ cannot be greater than 2ε, otherwise this would violate the condition that the open loop damping is d_o.

The facts that C_{gd} are non-empty and convex can now be used to more explicitly characterize its shape.

Extreme Points of C_{gd}

The sequel describes a method for determining the extreme points of C_{gd} along each of the axis in R^{m+1}, corresponding to each gain g_k, k=0, 1, ... m. In particular g_o (which is proportional to the closed loop damping through $g_o=2(\varepsilon+d-d_o)$), has a maximum value which will be evaluated. If this maximum value of g_o lies above 2ε then the open loop damping will have been improved. Furthermore, it will be shown that a convex set of g's εR^m can then be defined by joining the extreme points of C_{gd} yielding improved damping. Procedures for extending the damping beyond this maximum value are proposed later on.

Consider equation (9). The definiteness condition can be rephrased by the requirement that for all $x \varepsilon R^n$ the corresponding z_k, where

$$z_k = x^T J_k x \quad , \quad k = 0, 1, \ldots m \tag{16}$$

and

$$z = x^T x \tag{17}$$

satisfy

$$-z + \sum_{k=0}^{m} g_k z_k \leqslant 0 \tag{18}$$

The set C_{gd} can then be characterized by an infinity of half spaces in R^{m+1} as in (18) described by all feasible values of z and z_k, k=0, 1, ... m, as defined by (16) and (17). Each of the defining planes

$$-z + \sum_{k=0}^{m} g_k \, z_k = 0 \tag{19}$$

will intersect each of the m+1 axis in R^{m+1} at a point defined by

$$g_k = z/z_k = \frac{x^T x}{x^T J_k x} \tag{20}$$

Because of the quadratic nature of z and z_k, the intersection points z/z_k can be shown to have extreme values. To define these extreme points first consider the following eigenvalue problems for k=0, 1,... m

$$(\lambda_k I - J_k) \, v_k = 0 \tag{21}$$

Since the J_k are real and symmetric, the eigenvalues are all real.

Denote now the maximum and minimum λ_k for each J_k as λ_{kmax} and λ_{kmin} respectively.

Now from the theory of regular pencils (see (1), Vol. I, page 317, Theorem 10 and 13) the following lemma can be formulated:

Lemma 3

The extreme values of the ratios z_k/z are

$$\lambda_{kmin} \leq z_k/z \leq \lambda_{kmax} \tag{22}$$

Furthermore these extreme values are defined by the corresponding eigenvectors, that is,

$$\frac{z_k}{z} \leq \lambda_{kmax} = \frac{v^T_{kmax} \, J_k \, v_{kmax}}{v^T_{kmax} \, v^T_{kmax}} \tag{23}$$

$$\geq \lambda_{kmin} = \frac{v^T_{kmin} \, J_k \, v_{kmin}}{v^T_{kmin} \, v^T_{kmin}} \tag{24}$$

The extreme values of z_k/z can be used to define the extreme values of z/z_k which are the extreme intersection points of the planes defining C_{gd} with the g_k axis. Since these extreme intersection points are defined by feasible values of z and z_k (see (23) and (24)), they contain a set of points on the boundary of C_{gd}. The following theorem summarizes the above:

Theorem 2

The extreme values of the intersection points z/z_k are given by

$$1/\lambda_{kmax} \leq z/z_k \leq 1/\lambda_{kmin} \quad \begin{array}{l} \text{if } \lambda_{kmax} \geq \lambda_{kmin} > 0 \\ \text{if } \lambda_{kmin} \leq \lambda_{kmax} < 0 \\ \text{if } \lambda_{kmax} > 0 > \lambda_{kmin} \end{array} \tag{25}$$

$$z/z_k \leq 1/\lambda_{kmin} \quad \text{if } \lambda_{kmin} < \lambda_{kmax} = 0 \tag{26}$$

$$1/\lambda_{kmax} \leq z/z_k \quad \text{if } \lambda_{kmax} > \lambda_{kmin} = 0 \tag{27}$$

Some of the extreme values of (25) through (27) represent points on the boundary of C_{gd}. Specifically these are the points where the g_k axis intersect the boundary of C_{gd}. These boundary points, and the inequality constraints which they impose on the values of g_k along the g_k axis inside C_{gd} are:

$$1/\lambda_{kmin} \leqslant g_k \qquad \text{if } \lambda_{kmin} < \lambda_{kmax} \leqslant 0 \text{ i.e. } J_k \leqslant 0 \tag{28}$$

$$g_k \leqslant 1/\lambda_{kmax} \qquad \text{if } \lambda_{kmax} > \lambda_{kmin} \geqslant 0 \text{ i.e. } J_k \geqslant 0 \tag{29}$$

$$1/\lambda_{kmax} \leqslant g_k \leqslant 1/\lambda_{kmin} \qquad \text{if } \lambda_{kmax} > 0 > \lambda_{kmin} \tag{30}$$

Proof:

Equations (25) through (27) follow from Lemma 3. The fact that the extreme points lie on the boundary can be seen as follows. We have essentially searched the set C_{gd} along the directions defined by each g_k, setting all $g_j=0$, $j\neq k$. In this direction the set is characterized by the condition $z_k g_k \leqslant z$ for all feasible z and z_k. This condition implies

$$g_k \leqslant z/z_k \qquad \text{if } z_k > 0 \tag{31}$$

$$g_k \geqslant z/z_k \qquad \text{if } z_k < 0 \tag{32}$$

$$0 \leqslant z \qquad \text{if } z_k = 0 \tag{33}$$

Inequality (33) is immediately satisfied since $z=x^Tx$. From (32) the maximum value of z/z_k imposes the greatest restriction on g_k. Thus from (25) or (26),

$$g_k \geqslant \max z/z_k = 1/\lambda_{kmin} \tag{34}$$

Note that in the case shown by (27) where z/z_k has no maximum, $\lambda_{kmax} > \lambda_{kmin} = 0$. This then implies that $z_k \geqslant 0$ thus violating the assumption that $z_k < 0$ in (32). Since the extreme value of g_k shown in (34) is attainable, it must be a point on the boundary of C_{gd}.

In the last possibility given by (31), the minimum value of z/z_k imposes the greatest restriction on g_k. Thus from (25) and (27)

$$g_k \leqslant \min z/z_k = 1/\lambda_{kmax} \tag{35}$$

Again, in the case shown in (26) where z/z_k has no minimum, $\lambda_{kmin} < \lambda_{kmax} = 0$. This then implies that $z_k \leqslant 0$, thus violating the assumption that $z_k > 0$ in (31). Also since the extreme value of g_k shown in (35) is attainable, it must be a boundary point.

Finally in those cases where $z_k \geqslant 0$, for example $z_o=x^TMx$, it follows from (31) and (35) that only one of the two extreme points of z/z_k lies on the boundary of C_{gd}. This extreme point as given by (35) is $1/\lambda_{kmax}$ which is the smallest of the two.

The situation is similar when $z_k \leqslant 0$, except that the boundary point is then given by (34), that is $1/\lambda_{kmin}$.

This completes the proof of Theorem 2.

Maximum Damping in C_{gd}

In this subsection the maximum value of g_o (hence of d, the damping), in C_{gd} is determined by an iterative technique. This maximum value, g_o^* , is not the extreme value of g_o along the g_o axis only, as was the case in Theorem 2, but rather the absolute maximum over the entire set C_{gd}. The value of g_o^*

will then occur at some corresponding value of $g \neq 0$, g^* .

It will then follow, since C_{gd} is convex, that all the values of g_o and g defined by the line joining (g_o^*, g^*) to the boundary points of C_{gd} defined by equations (28) through (30) in Theorem 2, also lie in C_{gd}.

Furthermore it will be shown that an entire set contained in C_{gd} can be defined by a number of planes in R^{m+1}.

To obtain g_o^* consider the following algorithm.

Algorithm for Finding Maximum Damping in C_{gd}

a) Each boundary point of C_{gd} on the g_k axis defined in Theorem 2 defines an inequality condition of the type described in equation (18), where z and z_k, $k=0, 1, \dots m$, are defined by the eigenvector v_k corresponding to the eigenvalue λ_k (either λ_{kmax} or λ_{kmin}) defining each boundary point. The eigenvectors v_k are defined by equation (21). Thus for each of the r ($r \leqslant 2m+1$) boundary points

$$z \;=\; v_k^T v_k \;=\; 1 \tag{36}$$

and

$$z_k \;=\; v_k^T J_k v_k\,, \quad k=0, 1, \dots, m \tag{37}$$

Equation (36) is true if the v_k are normalized.

Defining now for the r ($r \leqslant 2m+1$) boundary points the following:

$$b_p \;=\; \begin{bmatrix} z_o \\ z_1 \\ \cdot \\ \cdot \\ \cdot \\ z_m \end{bmatrix}_p , \qquad p=1, 2, \dots r \tag{38}$$

corresponding to the p^{th} boundary point, the following inequality conditions must be satisfied by g_o and g,

$$b_p^T \begin{pmatrix} g_o \\ g \end{pmatrix} \leqslant 1 \;, \quad p = 1, 2, \dots, r \tag{39}$$

b) Solve the following linear programming problem:

$$\text{Maximize } g_o \text{ over } \begin{pmatrix} g_o \\ g \end{pmatrix} \in R^{m+1}$$

subject to the r inequality constraints defined by (39). A solution should exist since C_{gd} is non-empty. Call it g_k^* $k=0, 1, \dots m$.

In general this solution will not satisfy the stability condition, i.e. for all feasible z and z_k, $k=0, 1, \dots m$

$$-z + \sum_{k=0}^{m} g_k^* z_k \;\leqslant\; 0 \tag{40}$$

c) Verify the degree of "lack of stability" of this solution as follows:

$$\text{Maximize } x^T \left(-I + \sum_{k=0}^{m} g_k^* J_k\right) x \quad \text{over all } x \in R^n$$

This maximum value is given by the maximum eigenvalue, λ_{max}^*, of the matrix defining the above quadratic form. This means that for all $x \in R^n$,

$$x^T \left(-I + \sum_{k=0}^{m} g_k^* \, J_k \right) x \leq \lambda^*_{max} \, x^T x \tag{41}$$

If $\lambda^*_{max} \leq 0$ then we have found the point in C_{gd} of maximum damping, since (40) will be satisfied. If $\lambda^*_{max} > 0$, then its corresponding eigenvector, v^*_{max}, defines a new b_{r+1} where

$$b^T_{r+1} = (z_o, z_1 \ldots, z_m)_{r+1} \tag{42}$$

and

$$(z_k)_{r+1} = v^{*T}_{max} \, J_k \, v^*_{max}, \; k=0,\, 1,\, \ldots m \tag{43}$$

Clearly this value of b_{r+1} violates (40), the stability condition.

We now add the inequality

$$b^T_{r+1} \begin{pmatrix} g_o \\ g \end{pmatrix} \leq 1 \tag{44}$$

to those of (39), and repeat the process until λ^*_{max} is close to zero. The desired solution would be given by the corresponding value of g^*_k, $k=0, 1, \ldots m$. The value of g^*_o is the maximum $g_o \, \varepsilon \, C_{gd}$ and the value of g^* the corresponding gains.

This completes the algorithm for determining the maximum damping in C_{gd}.

Characterization of C_{gd}

The r boundary points defined in Theorem 2, together with the point corresponding to the maximum damping in C_{gd}, (g^*_o, g^*), serve to define a convex subset of C_{gd} whose maximum damping equals that of C_{gd}. To see this denote the point on the boundary of C_{gd}, y_o as

$$y_o = (\, g^*_o,\, g^* \,) \tag{45}$$

Also denote the r boundary points defined in Theorem 2 by y_j, $j=1, 2, \ldots r$. A typical value of y_j would be defined as follows:

$$y_j = (\, g^j_o \,,\, g^j \,) \tag{46}$$

where if y_j corresponds to a boundary point on the g_k axis of the type defined by the upper bound of (30) then

$$g^j_i = 1/\lambda_{kmin} \,, \qquad i = k \tag{47}$$

$$ = 0 \qquad\qquad i \neq k \tag{48}$$

By joining the r+1 boundary points y_j, $j=0, 1, \ldots r$ in groups of m+1 we can define ${}^{r+1}C_{m+1} \hat{=} S$ planes in R^{m+1} all of which must lie inside C_{gd} since this is a convex set.

These planes are found by straight forward (although non-trivial) methods in analytic geometry and are here denoted by

$$n^T_s \begin{pmatrix} g_o \\ g \end{pmatrix} = 1, \quad s = 1, 2, \ldots S = {}^{r+1}C_{m+1} \tag{49}$$

The vectors $n_s \, \varepsilon \, R^{m+1}$, $s=1,2,\ldots S$ can be found by the above mentioned methods.

The totality of these planes can serve to define a convex subset of C_{gd}, S_{gd}. This can be done by defining

$$S_{gd} = \left\{ \begin{pmatrix} g_o \\ g \end{pmatrix} / \; n_s^T \cdot \begin{pmatrix} g_o \\ g \end{pmatrix} \leqslant 1, \; s=1, 2, \ldots S \right\} \tag{50}$$

for the case where the zero component of n_s is greater than zero. If this component is zero then the plane is discarded. If this component is negative then the inequality sign in (50) is reversed.

The subset S_{gd} may contain redundant planes which make it smaller than necessary. Further studies are required to develop systematic methods of increasing S_{gd} by eliminating redundant planes.

ILLUSTRATIVE EXAMPLE

Consider the second order system with one feedback parameter g_1, where

$$A_o = \begin{bmatrix} 0 & 1 \\ 2 & -1 \end{bmatrix} \qquad \text{and} \qquad A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

The open loop damping is $d_o=-1$. Two cases are investigated one where $\varepsilon=0.1$ ("small") and one where $\varepsilon=1$ ("large"). In both figures the solid dots on the axis represent the corresponding boundary points of C_{gd}. The dotted lines represent the planes defining the various iterations, g_o^k, for the maximum damping. These are shown by triangles. The set S_{gd} is indicated by the area below the solid lines joining the last iteration to the boundary points on the axis. The largest S_{gd} is shown. The intersection of S_{gd} with the area above the solid horizontal line in Fig. 1 indicates the set for which damping is improved. In this case positive damping is not achieved, however in Fig. 2 the area above the second horizontal line indicates that positive damping is achievable. The solid curves are plots of the actual g_o versus gain, g_1. The technique therefore gives a conservative value of the achievable damping.

It can be noted that although the case where $\varepsilon=0.1$ leads to a lower maximum damping, it gives values of damping versus gains closer to the exact curve. To achieve higher damping with small ε, we might simply redefine A_o by $A(g^*)$, the value of $A(g)$ with the greatest damping, and repeat the process again.

CONCLUSIONS

A method has been described for constructing convex subsets of that region in the space of feedback gains leading to a certain system damping. This set is characterized by a number of linear inequality constraints in the gains. The results can be useful as a design tool in itself or as a complementary tool to more elaborate schemes. The desired scheme appears to be a sufficient but not necessary condition for improving the open loop damping via feedback. Further research is needed to strengthen the sufficient conditions.

REFERENCES

1. Gantmacher, F.R., "The Theory of Matrices," Vol. I & II, Chelsea Pub. Co., New York, 1960.
2. Davidson, E.J. & R. Chatterjee, "A Note on Pole Assignment in Linear Systems with Incomplete State Feedback," IEEE Trans. Automatic Control, Vol. AC-16, pp. 98-99, Feb. 1971.

3. Kimura, H., "Pole Assignment by Gain Output Feedback," *IEEE Trans. Automatic Control*, Vol. AC-20,pp.509-516, Aug.1975.
4. Jameson, A., "Optimization of Linear Systems of Constrained Configuration," *Int. J. Control*, Vol.11, No.3, pp. 409-421, 1970.
5. Kreisselmeier, G., "Stabilization of Linear Systems by Constant Output Feedback Using the Riccati Equation," *IEEE Trans. Automatic Control*, Vol. Ac-20, No. 4, pp. 556-557, Aug.1975.
6. Undrill, J.M., "Dynamic Stability Calculations for an Arbitrary Number of Interconnected Synchronous Machines," *IEEE Trans. Power Apparatus and Systems*,Vol. PAS-87, pp. 835-844, March 1968.

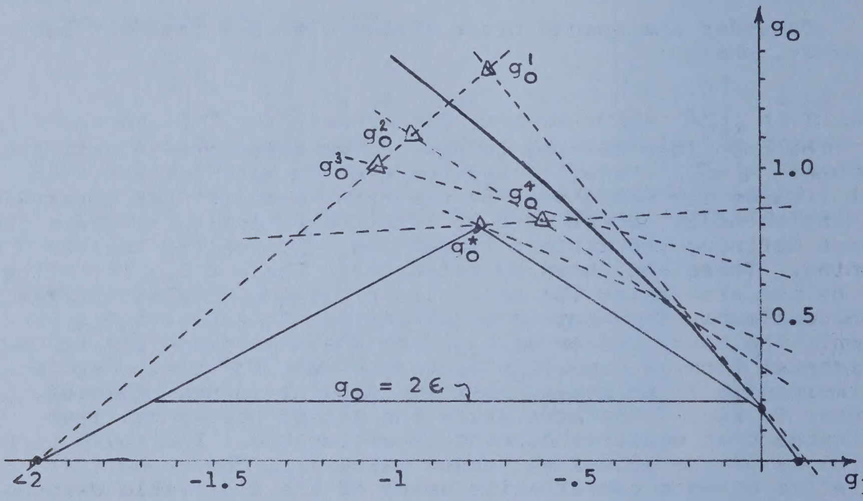


Figure 1: Second Order Example with $\epsilon = 0.1$

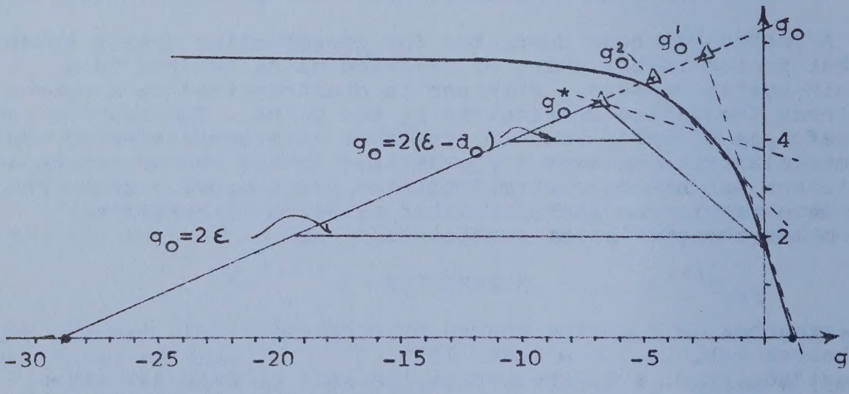


Figure 2: Second Order Example with $\epsilon = 1.0$.

A GRAPH ALGORITHM FOR THE OPTIMAL TEARING OF A SPARSE MATRIX

ALBERTO SANGIOVANNI-VINCENTELLI*
Dept. of Electrical Engineering
and Computer Sciences
University of California, Berkeley

ABSTRACT

Optimal tearing of a large sparse matrix A requires the selection of a non symmetric permutation, PAQ, such that PAQ is in an optimal bordered triangular form. A graph theoretic algorithm, based on bipartite graphs, is described and its properties are proved.

I. INTRODUCTION

Tearing is a particular strategy often used in large scale system analysis. Diakoptics is a classical example of application of this technique to electrical network analysis [1,2,3]. It consists mainly of two parts: at first, given a linear algebraic system of equations Ax=b, the solution of the system A*x=b is computed, A* being the matrix obtained by eliminating some elements of A, then this solution is modified to take into account the real structure of the original system. In general the structure of A* is fixed and, in order to optimize the tearing performance, it is necessary to minimize the "dimension"of the removed part. In particular, A* may be prescribed to be in inferior triangular form. Then an optimization problem arises: to reduce a given matrix $A \in \mathbb{R}^{n^2}$ via row and column permutations to a bordered triangular form (Fig. 1) in which A_{12} (the torn part) is of minimum rank. This optimization problem has been investigated by Cheung and Kuh [4,5] and Kevorkian [6] when only symmetric permutations on A are allowed. In [7], the author proposed a graph theoretic interpretation based on digraphs of the problem when nonsymmetric permutations are allowed. In [8], the author has introduced a new interpretation based on bipartite graphs. In this paper a graph algorithm based on the last interpretation is proposed and the correcteness of the algorithm as well as its complexity and proved.

II. PRELIMINARY DEFINITIONS AND REMARKS

A graph theoretical background is presented in this Section. All the undefined terms are to be understood according to Harary [9]. Let G = (X,U) (G = (X,E)) be a (di)graph with a set of vertices or nodes X and a set of (directed) edges or arcs $U = \{\{x_i,x_j\}|x_i,x_j \in X\}$ $(E = \{(x_i,x_j)|x_i,x_j \in X\})$. A simple (directed) path $\mu(x_i,x_j)$ of length ℓ is an ordered set of distinct vertices.

$$\mu(x_i,x_j) = \{p_o,p_1,\dots,p_\ell\}$$

such that

$$p_o = x_i, \; p_\ell = x_j \quad \{p_k,p_{k+1}\} \in U((p_k,p_{k+1}) \in E)$$

---

[*] On leave from Istituto di Elettrotecnica ed Elettronica, Politecnico di Milano - Italy.

$$k = 0,\dots,\ell-1$$

$$x_i \neq x_j$$

A *simple (directed) cycle* η of length ℓ is an ordered set of distinct vertices: $\eta = \{p_o,\dots,p_\ell\}$ such that $p_o = p_\ell = \bar{x}$, $\{p_k,p_{k+1}\} \in U((p_k,p_{k+1}) \in E)$ $k = 0,\dots,\ell-1$. The *section (di)graph* defined on a subset $Y \subset X$ is $G(Y) \triangleq (Y,U(Y))$ $(=(Y,E(Y)))$ where $U(Y) = \{\{x_i,x_j\} \in U | x_i,x_j \in Y\}$ $(E(Y) = \{(x_i,x_j) \in E | x_i,x_j \in \bar{Y}\})$.

A digraph $G = (X,E)$ is strongly connected, if $\forall x_i,x_j \in X$, $\exists$ $\mu_1(x_i,x_j)$ and $\mu_2(x_j,x_i)$.

The *strongly connected components* $G_i = (X_i,E_i)$, $i = 1,\dots,q$, of a digraph $G = (X,E)$ are the section graphs $G(X_i)$ such that $G_i(X_i,E_i)$ is strongly connected and $\forall X_j \subseteq X$, $X_j \supset X_i$, $G(X_j)$ is not strongly connected.

A *bipartite (di)graph* $B = (S,T,U)$ $(B = (S,T,E))$ is a (di)graph $B = (X,U)$ $(B = (X,E))$ such that $S \cup T = X$, $S \cap T = \emptyset$ and the section (di)graphs $B(S)$ and $B(T)$ are both vertex graphs. Given a bipartite (di)graphs $B = (S,T,U)$ $(B = (S,T,E))$ a *matching* I is a subset of $U(E)$ such that no two edges in I are incident to the same node. A node $x \in X$ is said *covered* if there is an edge in I that incides in it. A *complete matching* is a matching such that all the nodes of the bipartite (di)graph are covered. Given a bipartite graph and a matching I, an *alternating simple path* $\lambda(x_i,x_j)$ is a simple path such that if the edges $\{p_k,p_{k+1}\}$ with k even are in I, the edges with k odd are not in I or viceversa. An *alternating simple cycle* ρ is an alternating simple path $\lambda(x_i,x_j)$ in which $x_i = x_j$. Given a bipartite graph $B = (S,T,U)$ a *dumb bell* is a couple of nodes $d = [s,t]$ such that $s \in S$, $t \in T$ and $\{s,t\} \in U$. Given a bipartite graph B and a complete matching I, the *fundamental dumb bell set* $D(I)$ w.r.t. I is the set of dumb bells individuated by the complete matching I. Given a bipartite graph $B = (S,T,U)$ a set of dumb bells D and a dumb bell $d_i \in D$, $(d_i=[s_i,t_i])$ the set of dumb bells

$$S\text{-adj}(d_i) = \{d \in D,\ d = [s,t] | \{s,t_i\} \in U\}$$

is called the *S-adjacent to d_i dumb bell set*; the set of dumb bells

$$T\text{-adj}(d_i) = \{d \in D,\ d = [s,t] | \{s_i,t\} \in U\}$$

is called the T-adjacent to d_i dumb bell set; the *S-degree* of d_i, $Sd(d_i)$, is the cardinality of the set $S\text{-adj}(d_i)$, the *T-degree* of d_i, $Td(d_i)$ is the cardinality of the set $T\text{-adj}(d\ )$. A set of dumb bells D individuates the following node sets: $S(D) = \{s \in S | \exists d \in D,\ d = (\bar{s},t)\}$ $T(D) = \{\bar{t} \in T | \exists d \in D,\ d = (s,\bar{t})\}$, $X(D) = S(D) \cup T(D)$. Let B be a bipartite graph with a complete matching I, then a set of dumb bells F is an *essential dumb bell set* of B, if the section graph $B(Y)$, where $Y = X-X(F)$ admits a unique complete matching. An essential dumb bell set of minimum cardinality MF is called *minimum dumb bell essential set of B*.

III. THE OPTIMUM BORDERED TRIANGULAR FORM AND BIPARTITE GRAPHS

It has to be noted that the "structural information" of a matrix $A \in \mathbb{R}^{n^2}$ can be efficiently coded with a graph. In particular, given a matrix $A = |a_{ij}| \in \mathbb{R}^{n^2}$, a bipartite graph $B[A]$ can be associated to A as follows: $B[A] = (S,T,U)$, with $|S| = |T| = n$, and $\{s_i,t_j\} \in U$ iff (if and only if)

$a_{ij} \neq 0$; i, j = 1,...,n. Bipartite graphs are particularly suited when a "non symmetric permutation strategy" has to be chosen. In fact, a non symmetric permutation on A, PAQ, has as effect the reordering of the rows and of the columns of A. By definition of B[A] it follows that B[PAQ] is isomorphic to B[A]. Then the structural properties of B[A] are maintained

Using this representation of A, in [8], it has been proved that the determination of a minimum dumb-bell essential set leads to an optimal nonsymmetric permutation, PAQ, on A. In particular, the following Theorem holds:

Theorem 3.1. The selection of a nonsymmetric permutation on A, PAQ, such that PAQ is in an optimal bordered triangular form with A_{11} non singular and A_{22} of minimum dimension, is equivalent to the determination of a minimum dumb bell essential set MF in B[A].

The problem of determining an optimal bordered triangular form, is then reduced to the one of selecting a MF in B[A]. The MF problem can be considered as a covering problem and an efficient algorithm for this kind of problems is not available due to the intrinsic difficulty of the combinatorial problem. In general, branch and bound techniques are implemented [4], or suboptimal algorithms are devised [7]. In this paper the second way is followed, due to the simpler and more efficient performances of this kind of algorithms.

It has to be noted that the request of the existence of a complete matching in B[A] (Y) increases heavily the difficulty of devising an algorithm. On the other hand, this request cannot be removed, being related with the constraint of non zero elements on the main diagonal of A_{11}, which assures the "solvability" of the torm system.

IV. THE MINIMAL DUMB BELL ESSENTIAL SET ALGORITHM

As remarked in the previous section, an algorithm for the selection of a minimum dumb bell essential set is not computationally feasible due to the enormous amount of computation time and storage needed. Then, we are looking for an essential set with some optimal properties: a minimal essential dumb-bell set.

A minimal dumb bell essential set is a dumb bell essential set such that no proper subset of it is a dumb bell essential set for a given bipartite graph. Moreover, among all the possible minimal essential sets, we try to find one with the lowest possible cardinality.

In order to devise a heuristic procedure to achieve our goal, recall that if F is an essential dumb bell set, then B[A] (Y), where Y = X-X(F), admits a unique complete matching. Now let us consider the following theorem proved in [8].

Theorem 4.1. Let B be a bipartite graph with a complete matching I. I is unique iff it is possible to number the dumb bells d ∈ D(I) so that

$$d_j \in \text{S-adj}(d_i) \Rightarrow j > i \qquad (4.1)$$

Then, if we order the dumb bells in B[A](Y) according to Theorem 4.1, and we order the dumb bells in F as the last ones in B[A], the dumb bell d_1 in B[A] has in general small S-degree (often the minimum S degree w.r.t. the other dumb bells) and large T-degree (often the maximum T-degree w.r.t. the other dumb bells)[11]. The same result obviously holds for the graph $B[A](X-X(d_1))$ w.r.t. the dumb bells d_2. Moreover, the dumb-bells in F have large both the S-degree and T-degree.

We use these remarks in order to devise a sequential choice procedure. In fact, the large complexity of covering problems as well as other difficult combinatorial problems arises from the necessity of using back track procedures. Then in order to lower the computational complexity we have to renounce to the back track procedures and devise suboptimal sequential choices.

In this case, if we name $\bar{F}$ the set of dumb bells not in F, we try to assign dumb bells in $\bar{F}$ with small S-degree and large T-degree, and dumb bells in F with large S-degree and large T-degree.

Selected the choice rule, we have to face the complete matching problem, i.e., we have to assure that the choice done at a certain step will not affect the possibility of finding a complete matching in the remaining steps. The algorithm we propose uses the strong connectivity algorithm by Tarjan [10] to accomplish this task. We assume for sake of simplicity that at the beginning a complete matching I in B[A] is provided. Moreover, we direct the bipartite graph B[A] as follows:

(i) from T to S the edges $\in I$

(ii) from S to T all the other edges.

We will use as auxillary structures two stacks α and T'.

F - ALGORITHM

STEP 0. Set $B^* = B[A]$, $\alpha = \phi$, $T' = \phi$.

STEP 1. Apply the strong connectivity algorithm to B* and put the strong connected components of B* on the top of a stack α.

STEP 2. Pick up the graph on the top of α, $B_{\alpha 1}$ and set $B^* = B_{\alpha 1}$. If α is void, flag the dumb bells in B[A] directed from T to S with the t node in T' end, else continue.

STEP 3. Let t_1' be the first element in the stack T' belonging to B*. If there is no such element in T' continue else find a dumb bell in B*, $\bar{d} = [\bar{s},\bar{t}] = [\bar{s},t_1']$ with maximum S-degree. If some ties occur pick any of them. Flag it in B[A] and go to STEP 5.

STEP 4. Find the set of dumb bells in B* with minimum S-degree and between them the one with maximum T-degree. If more than one have the same "maximum" T-degree, we proceed as follows: let $d_1^o,\dots,d_k^o,\dots,d_m^o$ be the selected dumb bells. For each d_k^o, consider the dumb bells $d_{k,i}$ which have as S node s_k^o and compute the T-degree of each of them. Add the obtained numbers and associate the sum to $d_k^o \cdot \bar{d} = [\bar{s},\bar{t}]$ is the dumb bell with the maximum associated number. Put in T' the T-nodes connected with $\bar{s}$.

STEP 5. If $\bar{d}$ is oriented from T to S, set $B^* = B^*(X-X(\bar{d}))$ else find a path in B* from $\bar{t}$ to $\bar{s}$ $\mu(\bar{t},\bar{s})$ and reverse the orientation of the edges in the cycle obtained joining μ with $(\bar{s},\bar{t})$ in B* and B[A]. Set $B^* = B^*(X-X(\bar{d}))$ and go to STEP 1.

Now, we prove the following correctness Theorem:

Theorem 4.2. Let D* be the set of dumb bells oriented from T to S in B[A] at the end of the algorithm and $D^{*\prime} \subset D^*$ be the set of flagged dumb bells. D* is a fundamental dumb bell set and D*' is an essential set of B[A].

Proof. (i) D* is a fundamental dumb bell set. In order to prove the first part of the Theorem, we will use an inductive procedure. At the first stage of the algorithm the edges oriented from T to S constitute a complete matching by hypothesis and the corresponding dumb bells constitute a fundamental dumb bell set. Suppose that at the i-th stage, the edges oriented from T to S constitute a complete matching, we want to prove that at the i+1-th stage the same result holds. Passing from i to i+1, we must do STEP 5. In this STEP, if the chosen dumb bell is directed from T to S the directions of the edges do not change and, then, the complete matching property is still valid. If the direction of the edge is from S to T, we find a cycle and we reverse the direction of the edges. According to the following lemma, proved in [7],

Lemma 4.1. Given a bipartite graph B = (S,T,U) and a complete bipartite matching I_1, any other possible bipartite complete matching in B, I_h, can be obtained from I_1 as follows: individuate one or more disjoint simple alternating cycles (e.g.,m) ρ^j w.r.t. I_1. Let I_1' be the set of edges in I_1 but not incident in the nodes of the alternating cycles, and $\bar{I}$ the set of edges incident in the vertices of the alternating cycles but not in I_1. Then $I_h = I_1' \cup \bar{I}$.

and by the obvious correspondence between alternating cycles and directed cycles in B[A], we maintain a complete matching.

It remains to prove that it is always possible to find a cycle passing through the nodes of the chosen dumb bell. By definition of strong connected component of a graph and by STEP 1 this cycle can always be found and the first part of the theorem is proved.

(ii) According to STEPS 3 and 5, the dumb bells in D* -D*' can be ordered such that (4.1) holds. Then by Theorem 4.1 and by definition of essential dumb bell set, the proof is completed.

It has to be noted that there is no guarantee that F is a minimal dumb bell essential set. We can "refine" this set adding the following Refinement Step REF. STEP. Consider one at a time the dumb bells in D*', $d_i^{*\prime}$, and the bipartite directed graph $B[A](X-X(D^{*\prime}) \cup d_i^{*\prime})$. If the graph is acyclic then $D^{*\prime} = D^{*\prime} - d_i^{*\prime}$.

Now we want to prove the complexity of the overall F-algorithm.

Theorem 4.3. The complexity of F algorithm is $O(n^2)$, where n is the dimension of the matrix A.

Proof. At each stage of the algorithm, we have to

(1) Apply the strong connectivity algorithm.
(2) Compute the S-degree and the T-degree of dumb bells.
(3) Pick up elements which maximize or minimize S and/or T degree.
(4) find a path in B*

(1) is proved to be O(n) if the number of edges in the tested graph is proportional to n in [10]. It has to be noted that for a sparse matrix the number of elements are proportional to the dimension of the matrix.

(2) requires the spinning of the bipartite graph once for S nodes and once for T-nodes and then it is O(n).

(3) requires a number of comparison proportional to the number of elements compared. In the worst case we have to compare (STEP 4) 3 times n elements.

(4) requires the spanning of the graph and if we apply breadth first or depth search techniques, is O(n). Thus, the overall complexity for the F algorithm is $O(n^2)$. The refinement STEP requires to find a path between two nodes (O(n)) for each dumb bell in D*' and then is atmost $O(n^2)$.

It has to be noted that many redundancies can be avoided in the practical implementation of the algorithm, described in an informal way in order to allow the reader to fully understand the conceptual steps.

V. CONCLUDING REMARKS

In this paper an $O(n^2)$ algorithm for the selection of a "suboptimal" bordered triangular form of a sparse matrix via nonsymmetric permutations is given. The algorithm is essentially based on a topological structure associated to the matrix: a bipartite graph. In an earlier paper [8] the author has given a graph theoretic interpretation of this optimization problem and the algorithm described here is a straightforward consequence of the achieved theoretical results. It has to be noted that the algorithm is based on a heuristic procedure due to the high complexity of the optimization problem. The algorithm has been proved to be correct and to have $O(n^2)$ complexity.

Some further work is needed in order to improve the performances of the algorithm itself and to insert some reduction rules parallel to the reduction rules introduced in [4,5,6] for the symmetric permutation case. To emphasize this need, let us consider the bipartite graph of Fig. 2. A minimum essential set is $[s_4,t_2]$, while the proposed algorithm selects as essential set $[s_1,t_1]$, $[s_4,t_4]$. It is possible to show that if a particular reduction rule based on the degree of the dumb bells is inserted the algorithm can find the minimum essential set. This rule together with a more sophisticated algorithm will be discussed in a forthcoming paper.

REFERENCES

[1] F. H. Branin, "The Relation Between Kron's Method and the Classical Methods of Network Analysis," IRE WESCON Convention Record, part 2, 3-28, (1957).

[2] L. O. Chua and L. K. Chen, "Non Linear Diakoptics," Proc. 1975 IEEE Int. Symp. on Circuits and Systems, Boston, Mass., May 1975.

[3] F. H. Branin, "A Sparse Matrix Modification of Kron's Method of Piecewise Analysis," Proc. 1975 IEEE Int. Symp. on Circuits and Systems, Boston, Mass., May 1975.

[4] L. K. Cheung and E. S. Kuh, "The Bordered Triangular Matrix and Minimum Essential Set of a Digraph," IEEE Trans. on Circuits and Systems, vol. CAS 21, pp. 633-639, Sept. 1974.

[5] L. K. Cheung and E. S. Kuh, "Feedback Vertex and Edge Sets of a Digraph and Applications to Sparse Matrices," Memo ERL-M419, University of California, Berkeley, February 1974.

[6] A. Kevorkian and J. Snoek, "Decomposition in Large Scale Systems: Theory and Applications in Solving Large Sets of Nonlinear Simultaneous Equations," in D. M. Himmemblau Ed., Decomposition of Large Scale Problems. North Holland Publ. Comp., 1973.

[7] A. Sangiovanni-Vincentelli, "Nonsymmetrical Permutations and the Optimal Bordered Triangular Form of a Sparse Matrix," Proc. 12th Annual Allerton Conference on Circuit and System Theory, Monticello Illinois, Oct. 1974.

[8] A. Sangiovanni-Vincentelli, "Bipartite Graphs and Optimal Tearing of a Sparse Matrix," Proc. 1975 Midwest Symposium, Montreal, Aug. 1975.

[9] F. Harary, Graph Theory, Reading, Mass., Addison-Wesley, 1969.

[10] R. Tarjan, "Depth First Search and Linear Graph Algorithms," SIAM J. Comput., vol. 1, no. 2, pp. 146-160, June 1972.

[11] K. Bhar and B. Kinariwala, "Optimum and Suboptimal Methods for Permuting a Matrix into a Bordered Triangular Form," unpublished manuscript.

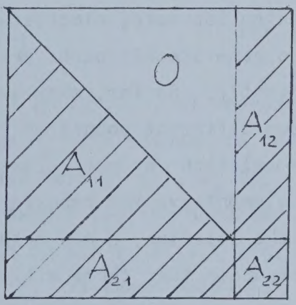


Fig 1

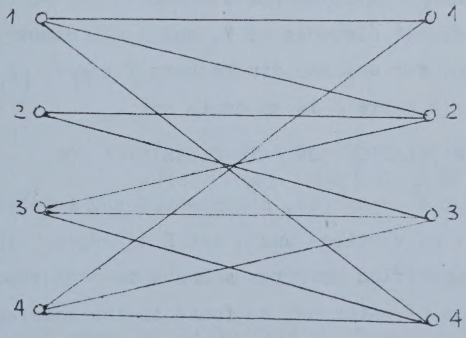


Fig.2

GENERATION OF PATHS BY ELEMENTARY TREE TRANSFORMATIONS

SHOJI SHINODA
Faculty of Science and
Engineering, Chuo University
Tokyo 112

WATARU MAYEDA
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

ABSTRACT

A new relation for classification of paths between two specified vertices in a connected, nonoriented and labeled graph is derived. In its derivation, the concept of elementary tree transformations plays a key rôle. Besides, by applying the relation, a new method of path generation is proposed.

INTRODUCTION

In connection with many practical applications in the fields of switching circuits, communication nets, electrical networks and so on, it is of importance for us to generate all paths between two specified vertices in a graph efficiently. So far, many available methods have been presented and improved from different points of view [1-4].

In this paper, a new relation for path classification is derived in a form that the concept of elementary tree transformations is explicitly used. Next, by applying the relation, a new method of path generation is proposed. The method guarantees that there will be no dulications in the generated paths.

Throughout this paper, the symbols " $\cup$ ", "$\cap$", and " - " are used to denote the set-theoretic binary operations called "union", "intersection", and "difference", respectively. For any finite set X, $|X|$ is used to denote the number of all elements of X, and $x \in X$ means that x is an element of X. Also, for any two finite sets X and Y, $X \cup Y$ is written as $X + Y$ if $X \cap Y = \phi$ where ϕ is an empty set.

A NEW RELATION FOR PATH CLASSIFICATION

Let us consider a connected, nonoriented and labeled graph G which consists of a set V of vertices and a set E of edges. The set of all paths between two specified vertices s and t in G is represented by P[G], where our definition of paths may be found in the references [1,2]. For an element P of P[G], a tree of G which has the element P as one of its tree-paths is denoted by T, where by a tree-path of a tree is meant a path made up only of some or all edges of the tree. For a subset E' of E such that $T \cap E' = \phi$ for the tree T, $G<T:E'>$ is used to denote the graph obtained from G by deleting all possible edges of E', and $P[G<T:E'>]$ is used to represent the set of all paths between two specified vertices s and t in $G<T:E'>$. Here, if $E' = \phi$, then

$$P[G] = P[G<T:E'>]. \tag{1}$$

Next, let us denote the set of all edges of the path P by $\{e_1, e_2, \dots, e_m\}$, and let us denote the fundamental cutsets defined by the edges $e_1, e_2, \dots$, and e_m with respect to the tree T by $S_{e_1}(T)$, $S_{e_2}(T)$, ... , and $S_{e_m}(T)$, respectively. From these m cutsets, furthermore, let us define new sets $E_{(k)}$ and $\bar{S}_{e_k}(T)$ $(k = 1, 2, \dots, m)$ such that

$$E_{(k)} = \begin{cases} \phi & \text{for } k = 1, \\ E_{(k-1)} \cup (\bar{S}_{e_{k-1}}(T) - \{e_{k-1}\}) & \text{for } 2 \leq k \leq m, \end{cases} \tag{2}$$

and

$$\bar{S}_{e_k}(T) = S_{e_k}(T) - E_{(k)} \qquad \text{for } 1 \leq k \leq m. \tag{3}$$

Then, each set $\bar{S}_{e_k}(T)$ $(k = 1, 2, \dots, m)$ represents the fundamental cutset defined by the edge e_k with respect to the tree T of $G<T:E_{(k)}>$. Also, these m newly defined sets $\bar{S}_{e_1}(T)$, $\bar{S}_{e_2}(T)$, ... , and $\bar{S}_{e_m}(T)$ are said to be ordered with respect to the sequence of edges $e_1, e_2, \dots, e_m$ of the path P if, for each of $k = 1, 2, \dots, m$, $\bar{S}_{e_k}(T)$ is such that for every element P' of $P[G<T:E_{(k)}>]$ there holds $|\bar{S}_{e_k}(T) \cap P'| = 1$. With the above definitions, the following theorem can now be obtained:

Theorem 1: For the sets $\bar{S}_{e_1}(T)$, $\bar{S}_{e_2}(T)$, ... , and $\bar{S}_{e_m}(T)$ which are ordered with respect to the sequence of edges $e_1, e_2, \dots, e_m$ of the path P, P[G] can be expanded into

$$\sum_{k=1}^{m} (|\bar{S}_{e_k}(T)| - 1) + 1$$

subsets as follows:

$$P[G] = \{P\} + \sum_{k=1}^{m} \Big(\sum_{e \,\varepsilon\, (\bar{S}_{e_k}(T) - \{e_k\})} P[G<T \cup \{e\} - \{e_k\}: E_{(K)} \cup (\bar{S}_{e_k}(T) - \{e\})>] \Big) \tag{4}$$

where $\sum_{k=1}^{m}$ means that the summation is over all k such that $|\bar{S}_{e_k}(T)| > 1$.

The proof of this theorem easily follows from the fact that (a) If $\bar{S}_{e_k}(T)$ $(k = 1, 2, \dots, m)$ satisfies $|\bar{S}_{e_k}(T) \cap P'| = 1$ for every elememt

P' in $P[G<T:E_{(k)}>]$, then

$$P[G<T:E_{(k)}>] = \sum_{e \in \bar{S}_{e_k}(T)} P[G<T \cup \{e\} - \{e_k\}:E_{(k)} \cup (\bar{S}_{e_k}(T) - \{e\})>],$$

and (b)

$$P[G<T:E_{(k)} \cup (\bar{S}_{e_k}(T) - \{e_k\})>] = \begin{cases} P[G<T:E_{(k+1)}>] & \text{for } 1 \leq k \leq m-1 \\ \{P\} & \text{for } k = m. \end{cases}$$

Therefore, the detailed description of the proof is omitted here.

Here it should be noted that, for each k such that $|\bar{S}_{e_k}(T)| > 1$, $P[G<T \cup \{e\} - \{e_k\}:E_{(k)} \cup (\bar{S}_{e_k}(T) - \{e\})>]$ in (4) not only represents a subset of P[G] whose elements contain the edges $e_1, e_2, \ldots, e_{k-1}$, e in common but do not contain any edge of $E_{(k)} \cup (\bar{S}_{e_k}(T) - \{e\})$, but also indicates that one of its elements can be characterized as a tree-path of a new tree $T \cup \{e\} - \{e_k\}$ obtained from the tree T by elementary tree transformation. Hence, as a consequence of the expansion of (4), we see that $\sum_{k=1}^{m} (|\bar{S}_{e_k}(T)|-1)$ elements in P[G] - {P} can be uniquely determined with respect to the path P as tree-paths of $\sum_{k=1}^{m} (|\bar{S}_{e_k}(T)|-1)$ trees obtained from the tree T by elementary tree transformations. This fact is noted from the view-point of proposing an efficient method of path generation.

A NEW METHOD OF PATH GENERATION

For any tree T that contain the path P as one of its tree-paths, there does not always exist a set of $\bar{S}_{e_1}(T), \bar{S}_{e_2}(T), \ldots, \bar{S}_{e_m}(T)$ which are ordered with respect to the sequence of edges $e_1, e_2, \ldots, e_m$ of the path P. However, for some tree T, there always exists such a set, and it is always possible to construct such a set. Intuitively, for example, if $\bar{S}_{e_k}(T)$ has any of graph configurations shown in Fig.1, then there always holds $|\bar{S}_{e_k}(T) \cap P'| = 1$ for every element P' in $P[G<T:E_{(k)}>]$. Such intuitive criteria are available to some extent for manual implementation but are not available for machine use. In this paper, by modifying the depth-first search on a graph [5], we give an available method of constructing a set of $\bar{S}_{e_1}(T), \bar{S}_{e_2}(T), \ldots, \bar{S}_{e_m}(T)$ which are ordered with respect to the sequence of edges $e_1, e_2, \ldots, e_m$ of a path P. The method has the advantage of being in a form suitable for machine use.

For convenience' sake, it is assumed that G does not contain any parallel edges. If an edge e of G is connected between two vertices u and v of G, then the edge e is denoted by e = (u,v) or e = (v,u). If we orient an edge $e = (u,v)$ of G from vertex u to vertex v, then the oriented edge e is represented by $e = \langle u,v \rangle$. Next, let us assume that each edge e_k ($k = 1, 2, \ldots, m$) of the path P is denoted by $e_k = (v_k, v_{k+1})$ where $v_1 = s$ and $v_{m+1} = t$, let all the vertices of G be unexplored, and let us consider the following search:

Step 1: Traverse an alternating sequence of vertices and edges $v_1, e_1, v_2, e_2, \ldots, v_m, e_m, v_{m+1}$ from the vertex $v_1 (= s)$ to the vertex $v_{m+1} (= t)$ of G. Set NUMBER(v_k) $\leftarrow k$ for all $k = 1, 2, \ldots, m$, $T \leftarrow \{e_1, e_2, \ldots, e_m\}$, $\bar{T} \leftarrow \phi$. Orient each edge $e_k = (v_k, v_{k+1})$ from vertex v_k to vertex v_{k+1} ($K = 1, \ldots, m$).

Step 2: Set $v \leftarrow v_{m+1}$, $i \leftarrow m+1$, NUMBER(v) $\leftarrow m+1$.

Step 3: Set $i \leftarrow i+1$.

Step 4: Look for an untraversed edge incident at vertex v.

(a) If there is no such edge, go to Step 6; otherwise,

(b) Pick the first untraversed edge at vertex v, say $e = (u,v)$ and traverse the edge. Now you are at vertex u. Go to Step 5.

Step 5: (a) If NUMBER(u) is undefined, orient the edge e from vertex u to vertex v and add the edge e to the set T. Set NUMBER(u) $\leftarrow i$. Set $v \leftarrow u$ and go to Step 3.

(b) If NUMBER(u) < NUMBER(v_{m+1}), orient the edge $e = (u,v)$ from vertex u to vertex v and add the edge e to the set $\bar{T}$. Go back to vertex v and go to Step 4.

(c) If NUMBER(u) $\geq$ NUMBER(v_{m+1}), orient the edge $e = (u,v)$ from vertex v to vertex u and add the edge e to the set $\bar{T}$. Go back to vertex v and go to Step 4.

Step 6: Check to see if there exists some traversed edge, say $e = (v,w)$, in T oriented away from vertex v.

(a) If there is such an edge, move back to vertex w through the edge e. Set $v \leftarrow w$ and go to Step 4.

(b) If there is no such edge, set $k \leftarrow m+1$ and go to Step 7. (Now you are at vertex v_{m+1}.).

Step 7: Set $k \leftarrow k - 1$. Set $v \leftarrow v_k$. Go to Step 9.

Step 8: Set $i \leftarrow i + 1$.

Step 9: Look for an untraversed edge incident at vertex v.

(a) If there is no such edge, go to Step 11; otherwise,

(b) Pick the first untraversed edge at vertex v, say $e = (u,v)$, and traverse the edge. Now you are at vertex u. Go to Step 10.

Step 10: (a) If NUMBER(u) is undefined, orient the edge $e = (u,v)$ from vertex u to vertex v and add the edge e to the set T. Set

NUMBER(u) $\leftarrow$ i. Set $v \leftarrow u$ and go to Step 8.

(b) If NUMBER(u) < NUMBER(v_k), orient the edge $e = (u,v)$ from vertex u to vertex v and add the edge e to the set $\bar{T}$. Go back to vertex v and go to Step 9.

(c) If NUMBER(u) $\geq$ NUMBER(v_k), orient the edge $e = (u,v)$ from vertex v to vertex u and add the edge e to the set $\bar{T}$. Go back to vertex v and go to Step 9.

Step 11: Check to see if K = 1.

(a) If $k > 1$, go to Step 7.

(b) If $k = 1$, stop.

Then, as a direct consequence of this search, we have the following:

(i) G is converted into an acyclic oriented (directed) graph $\vec{G}$.

(ii) T is a directed tree rooted at vertex $v_{m+1}(= t)$ of $\vec{G}$.

(iii) For each of $k = 1, 2, \ldots, m$, $\bar{S}_{e_k}(T)$ is the set of all edges incident at vertex v_k and oriented away from vertex v_k. Here it should be noted that $\bar{S}_{e_1}(T)$, $\bar{S}_{e_2}(T)$, ..., and $\bar{S}_{e_m}(T)$ are ordered with respect to the sequence of edges $e_1, e_2, \ldots, e_m$ of P.

(iv) A path of G, which is not only an element of $P[G<T \cup \{e\} - \{e_k\}: E_{(k)} \cup (\bar{S}_{e_k}(T) - \{e\})>]$ but also a tree-path of a tree $T \cup \{e\} - \{e_k\}$, is a directed path from vertex $v_1(= s)$ to vertex $v_{m+1}(= t)$ of $\vec{G}$ which consists of an oriented edge $e = <u,v>$, a directed tree-path from vertex v_1 to vertex u of a directed tree T and a directed tree-path from vertex v to vertex v_{m+1} of the directed tree T.

Example 1: Consider a graph G shown in Fig.2(a), where the set of all edges of a path P between two vertices s and t of G is given by $P = \{1, 2, 3\}$. Then, by our search algorithm just described, the order in which the edges are explored is given in Fig.2(b) and, for this order of traversal, $\vec{G}$ is given in Fig.2(c). Also, as a consequence of the search, $T = \{1, 2, 3, 4, 5\}$, $\bar{S}_1(T) = \{1, 6, 7\}$, $\bar{S}_2(T) = \{2, 8\}$ and $\bar{S}_3(T) = \phi$, where $e_1 = 1$, $e_2 = 2$, $e_3 = 3$ and $m = 3$. Hence the relation corresponding to (4) becomes

$$P[G] = \{P\} + P[G<T \cup \{6\} - \{1\}:\{1, 7\}>] + P[G<T \cup \{7\} - \{1\}:\{1, 6\}>] + P[G<T \cup \{8\} - \{2\}:\{2, 6, 7\}>]$$

and the elements of P[G] which are characterized as tree-paths of the trees $T \cup \{6\} - \{1\}$, $T \cup \{7\} - \{1\}$ and $T \cup \{8\} - \{2\}$ become $P_1 = \{6, 3\}$, $P_2 = \{7, 4, 2, 3\}$ and $P_3 = \{1, 8\}$, respectively.

The above search algorithm can also be applied to the case where G contains parallel edges.

As is easily seen from the above, we know that the repeated application of the expansion of (4) enables us to generate all the elements of P[G] without duplications. As a matter of fact, it is a method of generating all the elements of P[G] by the repeated application of the expansion of (4) that is our method of path generation.

CONCLUSIONS

We have given the theoretical basis for a new method of path generation.

In our search algorithm which plays a preprocessing part in application of the theorem 1, each vertex of G is numbered exactly once and each edge of G is traversed exactly twice. Therefore, the running time required by our search is linear in $|V| + |E|$. Also, the running time of determining each desired tree-path is proportional to $|V|$. Hence, roughly speaking from these facts, the running time required to carry out our method of path generation will be bounded by $|P[G]|(\alpha|V| + \beta|E|)$ for some constants α and β. Although it has been assumed in our method of path generation that one element of P[G] is known beforehand, if no element of P[G] is known beforehand, then we have only to construct any one element of P[G] by the depth-first search [5]. In the case, the running time required to construct it is proportinal to $|V| + |E|$. In a forthcoming paper, our method will be compared in detail with the methods of other authors from the viewpoint of practical machine implementation.

Our method of path generation is also to be noted in connection with the problem of tree generation [6].

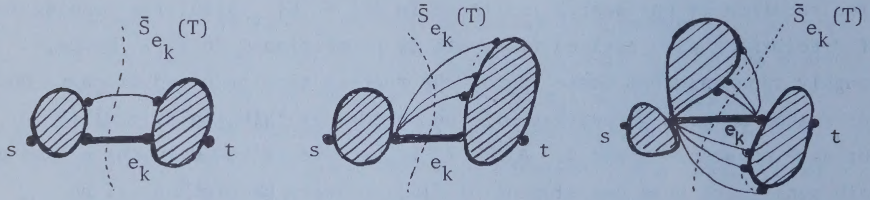
The dual treatment of the theoretical basis given here is expected in relation to cutset generation problem. However, it is not always possible in general because there does not always be any path P' of P[G] such that there holds $|P' \cap S| = 1$ for every cutset S separating two specified vertices s and t in G. But, if G belongs to some restricted class of graphs, then the theoretical basis given here can be dually applied to the problem of generating all cutsets separating two specified vertices s and t in G without duplications.

REFERENCES

1. W. Mayeda,"Graph Theory," John Wiley & Sons, New York, 1972.
2. S. Seshu and M. B. Reed,"Linear Graphs and Electrical Networks,"

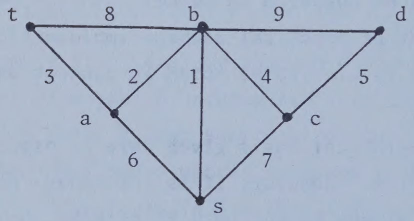
Addison-Wesley, Reading, Mass., 1961.

3. N. Deo,"Graph Theory with Applications to Engineering and Computer Science," Prentice-Hall, Englewood Cliffs, N.J., 1974. (Chapter 11: Graph-theoretic algorithms and computer programs).
4. E. Tanaka,"Diakoptical formulae for calculating paths and cuts," IECE Proceedings of Japan, 50, 1533-1540, 1967.
5. R. Tarjan,"Depth-first search and linear graph algorithms," SIAM J. Comput., 2, 146-160, 1972.
6. S. Shinoda,"On generation of trees and directed trees by shortcircuiting of branches," IECE Trans. of Japan, 53-A, 378-379, 1970.
7. S. Shinoda and R. Yatsuboshi,"Ariyoshi's relation for path classification and its extension," IECE Trans. of Japan, 58-A, 469-471, 1975.



(a) (b) (c)

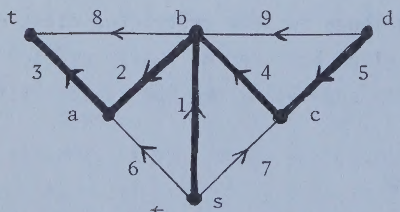
Fig.1 Examples of $\bar{S}_{e_k}$ (T) in G<T:$E_{(k)}$>



(a) Graph G before our search

1 = <s,b> ε T
2 = <b,a> ε T
3 = <a,t> ε T
8 = <b,t> ε $\bar{T}$
6 = <s,a> ε $\bar{T}$
4 = <c,b> ε T
7 = <s,c> ε $\bar{T}$
5 = <d,c> ε T
9 = <d,b> ε $\bar{T}$

(b) Order in which edges were explored



(c) Graph $\overleftarrow{G}$ after our search

Fig.2 Graphs for Example 1

SYSTEMATIZATION OF A NEW METHOD FOR GENERATION OF ALL PATHS

JOSE E. RUBINSTEIN
Instituto Costarricense de Electricidad, San José, Costa Rica

ABSTRACT

A new method for generating all possible paths has been previously introduced, which also generates all circuits in its first steps. Here, a way suitable for computer analysis is suggested for application of this method, using connection matrices. Once a solution is known a matrix can be formed to find other solutions, so that all paths can be generated for any pair of vertices in a graph without any duplications. A special case in oriented graphs yields up to four different sets of possible paths from a known solution.

Introduction

A new method has been proposed for generation of all possible paths between two vertices in a graph, which also generates all directed circuits while obtaining all possible paths [1]. This method may not be the fastest one at a first instance, but once a solution is obtained it becomes the fastest method available to obtain $\{P_{ik}\}$ knowing $\{P_{ij}\}$.

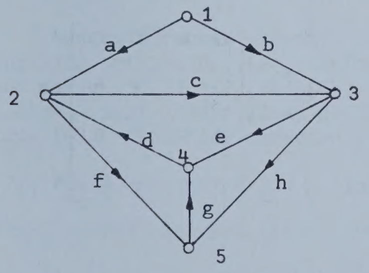
Here, an appropiate method is given for computerized applications, using a standard connection matrix.

Connection Matrix M

Let G be an oriented graph with n vertices. A connection matrix $C=\left[c_{ij}\right]$ of order n by n is defined by [2]

$$c_{ij}=\begin{cases} e & \text{if edge e is connected from vertex i to vertex j} \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

This graph may have self-loops or parallel edges with the same orientation. A connection matrix M and its corresponding linear graph are shown in Fig. 1.



(a)

$$M = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & a & b & 0 & 0 \\ 2 & 0 & 0 & c & 0 & f \\ 3 & 0 & 0 & 0 & e & h \\ 4 & 0 & d & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & g & 0 \end{array}$$

(b)

Fig. 1. (a) An oriented graph G; (b) corresponding connection matrix M.

Deletion of Vertices (Path Equivalent Reduction of Vertices)

In order to generate all possible paths between two vertices, remaining vertices must be deleted.

It is possible to assign to each edge a weight of the form

$$W(w) = \left\{ W_{pq}(w_{pq}) \,\middle|\, p,q = 1,2,\dots,m \right\} . \tag{2}$$

When $w_{pq} = \emptyset$, w_{pq} can be omitted. When $W_{pq} = \emptyset$, $W_{pq}(w_{pq})$ will be omitted.

To delete vertex k, it is convenient to first eliminate entry E_{kk} in M to avoid unnecessary operations. Consider column C_k and row R_k of connection matrix M, where

$C_k = \left[D_{yk} \ (d_{yk}) \right]_{p \cdot 1}$ and $R_k = \left[F_{ky} \ (f_{ky}) \right]_{1 \cdot p}$, $y \neq k$.

Let us use the weight product [1] to obtain a new matrix A_k, such that

$$A_k = C_k \overset{*}{k} R_k = \left[D_{yk}(d_{yk}) \overset{*}{k} F_{ky}(f_{ky}) = \begin{cases} D_{yk} U F_{ky} (d_{yk} U f_{ky} U k) \\ \qquad \text{for } d_{yk} \cap d_{yk} = \emptyset \\ \emptyset \quad \text{for } d_{yk} \cap d_{ky} \neq \emptyset \end{cases} \right]_{p \cdot p} . \tag{3}$$

We can eliminate row R_k and column C_k of connection matrix M, adding the resultant matrix M' to A_k. The new connection matrix $M_k = A_k U M'$ corresponds to oriented graph G with vertex k deleted.

Example 1

In order to delete vertex 2 from oriented graph G in Fig. 1. (a), first obtain connection matrix A_2:

$$A_2 = \begin{array}{c} \\ 1 \\ 3 \\ 4 \\ 5 \end{array} \!\!\begin{array}{c} 2 \\ \left[\begin{array}{c} a \\ 0 \\ d \\ 0 \end{array}\right] \end{array} \quad \overset{*}{2} \quad \begin{array}{cc} & \begin{array}{cccc} 1 & 3 & 4 & 5 \end{array} \\ 2 & \left[\begin{array}{cccc} 0 & c & 0 & f \end{array}\right] \end{array} = \begin{array}{c|cccc} & 1 & 3 & 4 & 5 \\ \hline 1 & 0 & ac(2) & 0 & af(2) \\ 3 & 0 & 0 & 0 & 0 \\ 4 & 0 & cd(2) & 0 & df(2) \\ 5 & 0 & 0 & 0 & 0 \end{array}$$

Then

$$A_2 U M' = \begin{array}{c|cccc} & 1 & 3 & 4 & 5 \\ \hline 1 & 0 & ac(2) & 0 & af(2) \\ 3 & 0 & 0 & 0 & 0 \\ 4 & 0 & cd(2) & 0 & df(2) \\ 5 & 0 & 0 & 0 & 0 \end{array} \quad + \quad \begin{array}{c|cccc} & 1 & 3 & 4 & 5 \\ \hline 1 & 0 & b & 0 & 0 \\ 3 & 0 & 0 & e & h \\ 4 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & g & 0 \end{array}$$

and

$$M_2 = A_2UM' = \begin{array}{c|cccc} & 1 & 3 & 4 & 5 \\ \hline 1 & 0 & b,ac(2) & 0 & af(2) \\ 3 & 0 & 0 & e & h \\ 4 & 0 & cd(2) & 0 & df(2) \\ 5 & 0 & 0 & g & 0 \end{array}$$

Connection matrix M_2 corresponds to oriented graph G_1 in Fig. 2.

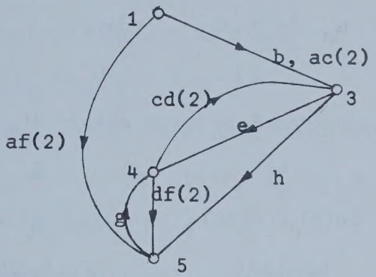


Fig. 2. Oriented graph G_1, produced by path equivalent reduction of vertex 2 in graph G.

Deletion of vertices must be performed as many times as necessary to reduce connection matrix M_{k_1} to the order 2 by 2. The end result will be of the form

$$M_{k_1 k_2 \dots k_{n-2}} = \begin{array}{c|cc} & i & j \\ \hline i & D_{ii}\ (d_{ii}) & P_{ij}(P_{ij}) \\ j & P_{ji}\ (P_{ji}) & D_{jj}(d_{jj}) \end{array} . \qquad (4)$$

Directed Circuits

While obtaining connection matrix A_k it was pointed out that it is convenient to first eliminate entry E_{kk}. This avoids unnecessary operations, for row k and column k are not needed in A_k. However, entry E_{kk} corresponds to self-loops in linear graph G, which are directed circuits [1].

Hence, directed circuits can be obtained during deletion of vertices, checking entries D_{kk} (d_{kk}) before eliminating them to obtain matrices A_k, and making $C(c) = D_{kk}\ (d_{kk}Uk)$. Notice that k = i, j must be considered to include self- loops given by $D_{ii}(d_{ii})$ and $D_{jj}(d_{jj})$ in final connection matrix (4). Remaining directed circuits are obtained from

$\{P_{ij}(P_{ij})\} \overset{*}{ij} \{P_{ji}(P_{ji})\}$ in (4), where the weight product is the one defined in (3) but with k=ij. Thus,

$$\{C(c)\} = \{D_{kk}(d_{kk}Uk) \mid k=k_1, k_2 \dots, k_{n-2}, i,j\}$$
$$U\ (\{P_{ij}(p_{ij})\} \overset{*}{ij} \{P_{ji}\ (p_{ji})\}) \qquad (5)$$

Example 2

Consider deletion of vertex 3 in oriented graph G_1 in Fig. 2. This corresponds to elimination of vertex 3 in connection matrix M_2.

Entry $D_{33}(d_{33}) = 0$, so there is no subset of directed circuits for this reduction. Connection matrix A_3 is given by

$$A_3 = \begin{array}{c|c|} & 3 \\ \hline 1 & b,ac(2) \\ 4 & cd(2) \\ 5 & 0 \\ \hline \end{array} \underset{3}{*} \begin{array}{c|ccc|} & 1 & 4 & 5 \\ \hline 3 & 0 & e & h \\ \hline \end{array} = \begin{array}{c|ccc|} & 1 & 4 & 5 \\ \hline 1 & 0 & be(3),ace(23) & bh(3),ach(23) \\ 4 & 0 & cde(23) & cdh(23) \\ 5 & 0 & 0 & 0 \\ \hline \end{array}$$

Deleting row 3 and column 3 in M_2 we obtain M_2', and

$$M_{23} = A_3 U M_2' = \begin{array}{c|ccc|} & 1 & 4 & 5 \\ \hline 1 & 0 & be(3),ace(23) & af(2),bh(3),ach(23) \\ 4 & 0 & cde(23) & df(2),cdh(23) \\ 5 & 0 & g & 0 \\ \hline \end{array}$$

Fig. 3 (a) shows the oriented graph defined by M_{23}. At this step we can obtain $\{P_{14}\}$ and $\{P_{41}\}$ by deleting vertex 5, $\{P_{45}\}$ and $\{P_{54}\}$ by reducing vertex 1, as well as $\{P_{15}\}$ and $\{P_{51}\}$ by deleting vertex 4. Now $D_{44}(d_{44})$= cde(23), so $C_1(c_1) = D_{44}(d_{44}U4)$= cde(234).

$$A_4 = \begin{array}{c|c|} & 4 \\ \hline 1 & be(3),ace(23) \\ 5 & g \\ \hline \end{array} \quad \underset{4}{*} \quad \begin{array}{c|cc|} & 1 & 5 \\ \hline 4 & 0 & df(2),cdh(23) \\ \hline \end{array}$$

$$= \begin{array}{c|cc|} & 1 & 5 \\ \hline 1 & 0 & bdef(234) \\ 5 & 0 & dfg(24),cdgh(234) \\ \hline \end{array}$$

$$M_{234} = A_4 U M'_{23} = \begin{array}{c|cc|} & 1 & 5 \\ \hline 1 & 0 & af(2),bh(3),ach(23),bdef(234) \\ 5 & 0 & dfg(24),cdgh(234) \\ \hline \end{array} \qquad (6)$$

Fig. 3 (b) shows the oriented graph defined by M_{234}.

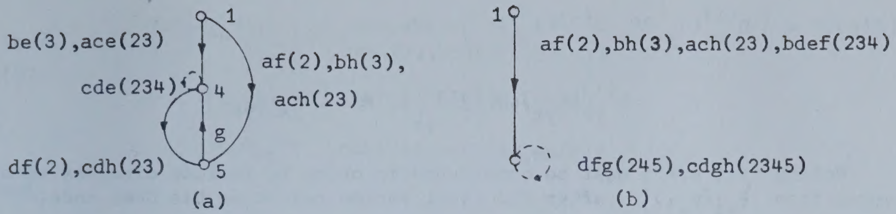


Fig. 3. (a) Oriented graph G_2, produced by path equivalent reduction of vertex 3 in G_1. (b) Oriented graph G_3, obtained after deletion of vertex 4 in oriented graph G_2.

From connection matrix M_{234} it is clear that

$$\begin{aligned} \{P_{51}(p_{51})\} &= \emptyset \\ \{P_{15}(p_{15})\} &= \{\text{af}(2),\text{bh}(3),\text{ach}(23),\text{bdef}(234)\}. \end{aligned} \tag{7}$$

Also, from $D_{55}(d_{55}), C_2(c_2) = \text{dfg}\ (245)$, $C_3(c_3) = \text{cdgh}(2345)$,

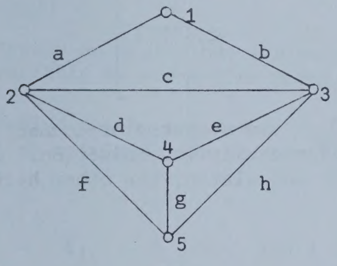
and $\{P_{15}(p_{15})\}\ 1\overset{*}{5}\ \{P_{51}(p_{51})\} = \emptyset$.

Thus $\{C(c)\} = \{\text{cde}(234),\text{dfg}(245),\text{cdgh}(2345)\}$ (8)

Generation of $\{P_{ij}\}$ for non-oriented graphs.

A connection matrix M can be determined for non-oriented graphs considering each non-oriented edge as two parallel oriented edges with opposite directions.

As an example, Fig. 4 shows a non-oriented graph and its corresponding connection matrix.



(a)

$$M = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & a & b & 0 & 0 \\ 2 & a & 0 & c & d & f \\ 3 & b & c & 0 & e & h \\ 4 & 0 & d & e & 0 & g \\ 5 & 0 & f & h & g & 0 \end{array}$$

(b)

Fig. 4. (a) Non-oriented graph G. (b) Corresponding connection matrix M.

The procedures outlined for oriented graphs are applicable to non-oriented graphs, with the exceptions that entries $D_{kk}(d_{kk})$ are no longer circuits. Instead, circuits are obtained from entries in columns $C_k = \left[D_{yk}(d_{yk})\right]_{p\times 1}$, $y \neq k$, when forming connection matrix A_k and after the final vertex reduction as

$$\{C(c)\} = \{ D^{(1)}_{yk}(d_{yk}) \ast_{yk} D^{(2)}(d_{yk}) \mid y \neq k, k = k_1, k_2, \ldots, k_{n-2}, j,$$

$$D^{(1)}_{yk}(d_{yk}), D^{(2)}_{yk}(d_{yk}) \in \{D_{yk}(d_{yk})\} \} . \qquad (9)$$

Notice that k = j must be considered in order to include circuits to be formed from $\{P_{ij}(p_{ij})\}$ after the final vertex reduction has been made.

This procedure does not duplicate circuits [1] .

Example 3

To delete vertex 2 in connection matrix M in Fig. 4(b), the following procedure is used. Entries $D_{kk}(d_{kk})$ are set to zero for convenience.

$$A_2 = \begin{array}{c} \\ 1 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} 2 \\ \left[\begin{array}{c} a \\ c \\ d \\ f \end{array}\right] \end{array} \overset{*}{2} \; 2 \begin{array}{c} \begin{array}{cccc} 1 & 3 & 4 & 5 \end{array} \\ \left[\begin{array}{cccc} a & c & d & f \end{array}\right] \end{array} = \begin{array}{c|cccc} & 1 & 3 & 4 & 5 \\ \hline 1 & 0 & ac(2) & ad(2) & af(2) \\ 3 & ac(2) & 0 & cd(2) & af(2) \\ 4 & ad(2) & cd(2) & 0 & df(2) \\ 5 & af(2) & cf(2) & df(2) & 0 \end{array}$$

Thus

$$M_2 = A_2UM' = \begin{array}{c|cccc} & 1 & 3 & 4 & 5 \\ \hline 1 & 0 & b,ac(2) & ad(2) & af(2) \\ 3 & b,ac(2) & 0 & e,cd(2) & h,cf(2) \\ 4 & ad(2) & e,cd(2) & 0 & g,df(2) \\ 5 & af(2) & h,cf(2) & g,df(2) & 0 \end{array}$$

No circuits are obtained from column C_2. The symmetry of connection matrix M for non-oriented graphs can be of advantage in its reduction. Only the upper triangle of the matrix needs to be calculated; the other half is symmetric.

$$A_3 = \begin{array}{c} \\ 1 \\ 4 \\ 5 \end{array} \begin{array}{c} 3 \\ \left[\begin{array}{c} b.ac(2) \\ e,cd(2) \\ h,cf(2) \end{array}\right] \end{array} \overset{*}{3} \; 3 \begin{array}{c} \begin{array}{ccc} 1 & 4 & 5 \end{array} \\ \left[\begin{array}{ccc} b,ac(2) & e,cd(2) & h,cf(2) \end{array}\right] \end{array}$$

$$= \begin{array}{c|ccc} & 1 & 4 & 5 \\ \hline 1 & 0 & E_{14} & E_{15} \\ 4 & E_{14} & 0 & E_{45} \\ 5 & E_{15} & E_{45} & 0 \end{array} .$$

Where
$$E_{14} = \{ace(23), bcd(23), be(3)\}$$
$$E_{15} = \{ach(23), bcf(23), bh(3)\}$$
$$E_{45} = \{cdh(23), cef(23), eh(3)\},$$

and

$$M_{23} = A_3 U M_2' = \begin{array}{c|ccc} & 1 & 4 & 5 \\ \hline 1 & 0 & E_{14}' = ad(2), E_{14} & E_{15}' = af(2), E_{15} \\ 4 & E_{14}' & 0 & E_{45}' = g, df(2), E_{45} \\ 5 & E_{15}' & E_{45}' & 0 \end{array} .$$

A subset of circuits $\{C_1(c_1)\} \subset \{C(c)\}$ is obtained from column C_3 using (9),

$$\{C_1(c_1)\} = \{abc(123), cde(234), cfh(235)\} .$$

Using M_{23} we can obtain $\{P_{14}\}$, $\{P_{15}\}$, and $\{P_{45}\}$, deleting the appropiate vertex each time, and the complete set of circuits using (9)with column C_k and with $\{P_{ij}\}$, for i,k,j,=1,4,5 depending on choice of $\{P_{ij}\}$.

Then $\{C(c)\} = \{C_1(c_1)\} \cup \{C_2(c_2)\} \cup \{C_3(c_3)\}$.

Generation of $\{P_{ij}(p_{ij})\}$ for all i and j.

The advantage of using this method is that once a solution $\{P_{ij}(p_{ij})\}$, $\{C(c)\}$ is known, it becomes the fastest method available to obtain $\{P_{ik}(p_{ik})\}$.

Assume an edge e between vertices k and j is known from the information available in connection matrix M. If $\{C(c)\}$ and $\{P_{ij}(p_{ij})\}$ are available, including the case when $\{P_{ij}\} = \emptyset$ and/or $\{C(c)\} = \emptyset$, then a 3 by 3 connection matrix M_p can be constructed as shown:

$$M_p = \begin{array}{c|ccc} & i & k & j \\ \hline i & 0 & E_{ik} = \{P_{ij} - e(p_{ij} - k) \mid e \in P_{ij}\} & E_{ij} = \{P_{ij}(p_{ij}) \mid k \notin p_{ij}\} \\ k & E_{ik} & 0 & E_{kj} = \{e, C - e(c - kj) \mid e \in C, kj \in c, i \notin c\} \\ j & E_{ij} & E_{kj} & 0 \end{array} \quad (10)$$

where $P_{ij}(p_{ij}) \in \{P_{ij}(p_{ij})\}$, and $C(c) \in \{C(c)\}$. (11)

The graph corresponding to M_p is shown in Fig. 5(a). From M_p we can obtain $\{P_{ik}(p_{ik})\}$ deleting vertex j, and $\{P_{jk}(p_{jk})\}$ deleting vertex i.

Similarly, knowledge of $\{P_{ij}(p_{ij})\}$, $\{C(c)\}$ and an edge e from vertex k to vertex j in an oriented graph, allow us to obtain all possible paths between i and k. Only $\{P_{ik}(p_{ik})\}$ is guaranteed by this method [1], so entries E_{ki}, E_{kj} and E_{ji} are set to zero in M_p:

$$M_p = \begin{array}{c} \\ i \\ k \\ j \end{array} \begin{array}{c} \begin{array}{ccc} i & k & j \end{array} \\ \left[\begin{array}{ccc} 0 & \{P_{ij}-e(p_{ij}-k) \mid e \in P_{ij}\} & \{P_{ij}(p_{ij}) \mid k \notin p_{ij}\} \\ 0 & 0 & 0 \\ 0 & \{C-e(c-kj) \mid e \in C, kj \in c, i \notin c\} & 0 \end{array}\right] \end{array} \tag{12}$$

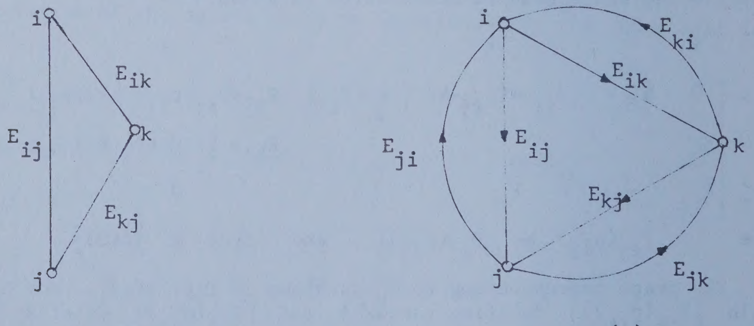
where (11) also applies. Reducing vertex j we get $\{P_{ik}(p_{ik})\}$, which is the only information obtainable from connection matrix M_p.

A more interesting result is obtained when an edge g from vertex i to vertex k is known besides edge e from k to j. Then we can use $\{P_{ij}(p_{ij})\}$, $\{P_{ji}(p_{ji})\}$ and $\{C(c)\}$ to construct a connection matrix M_p in the form

$$M_p = \begin{array}{c} \\ i \\ k \\ j \end{array} \begin{array}{c} \begin{array}{ccc} i & k & j \end{array} \\ \left[\begin{array}{ccc} 0 & \left\{\begin{array}{l} P_{ij}-e(p_{ij}-k) \mid \\ e \in P_{ij}, k \in p_{ij} \end{array}\right\} & \left\{\begin{array}{l} P_{ij}(p_{ij}) \mid \\ k \notin p_{ij} \end{array}\right\} \\ \left\{\begin{array}{l} C-g(c-ik) \mid \\ g \in C, ik \in c, j \notin c \end{array}\right\} & 0 & \left\{\begin{array}{l} P_{ij}-g(p_{ij}-j) \mid \\ g \in P_{ij}, j \in p_{ij} \end{array}\right\} \\ \left\{\begin{array}{l} P_{ji}(p_{ji}) \mid \\ k \notin p_{ji} \end{array}\right\} & \left\{\begin{array}{l} C-e(c-kj) \mid \\ e \in C, kj \in c, i \notin c \end{array}\right\} & 0 \end{array}\right] \end{array} \tag{13}$$

where $P_{ji}(p_{ji}) \in \{P_{ji}(p_{ji})\}$, and (11) applies.

The oriented graph corresponding to M_p is shown in Fig. 5(b). From connection matrix M_p we can not only obtain $\{P_{ik}\}$ and $\{P_{ki}\}$ by deleting vertex j, but also $\{P_{jk}\}$ and $\{P_{kj}\}$ by deleting vertex i.



(a) (b)

Fig. 5. (a) Graph corresponding to connection matrix M_p in (10). (b) Oriented graph corresponding to connection matrix M_p in (13).

Example 4

Suppose $\{P_{34}\}$ and $\{P_{43}\}$ have been determined from connection matrix M in Fig. 1(b) as

$$M_{125} = \begin{array}{c} \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cc} 3 & 4 \end{array} \\ \left[\begin{array}{cc} 0 & e,hg(5) \\ cd(2) & dfg(25) \end{array}\right] \end{array}$$

with $\{C(c)\}$ given by (8). Should we wanted to find $\{P_{42}\}$ we would not be able to use other simple methods [1] for there is no edge from vertices 2 to 4, as determined from connection matrix M in Fig. 1(b). However, not only edge d is from vertex 4 to vertex 2, but an edge c is from vertex 2 to vertex 3. We can then use (13) to construct

$$M_p = \begin{array}{c|ccc} & 4 & 2 & 3 \\ \hline 4 & 0 & d & 0 \\ 2 & 0 & 0 & c \\ 3 & e,\ hg(5) & 0 & 0 \end{array} .$$

Then

$$M_{p3} = \begin{array}{c|cc} & 4 & 2 \\ \hline 4 & 0 & \{P_{42}\} \\ 2 & \{P_{24}\} & 0 \end{array} = \begin{array}{c|cc} & 4 & 2 \\ \hline 4 & 0 & d \\ 2 & ce(3),cgh(35) & 0 \end{array}$$

and

$$M_{p4} = \begin{array}{c|cc} & 2 & 3 \\ \hline 2 & 0 & \{P_{23}\} \\ 3 & \{P_{32}\} & 0 \end{array} = \begin{array}{c|cc} & 2 & 3 \\ \hline 2 & 0 & c \\ 3 & de(4),dgh(45) & 0 \end{array}$$

can be obtained without difficulty.

Conclusion

A systematic way to apply a new method for generation of all possible paths in a given graph has been shown. This method is now suitable for computerized applications. Once a solution is known, it is enough to know an edge connecting another vertex to one of the vertices already solved to be able to find another solution. All circuits in the graph are obtained after the first final reduction, independently of the vertices used, and without duplication.

This procedure enable us to find all possible paths between any two vertices for a non-oriented graph, once a first solution is obtained.

For oriented graphs it may be necessary to use the complete reduction if a connecting edge is not known, but there is a special case which allows us to obtain four sets of all possible paths from a known solution.

References

[1] W. Mayeda, "Generation of all paths", to be published in IEEE Trans. in Circuits and Systems.

[2] Mayeda, W.. Graph Theory, John Wiley, 1972.

A SHORTEST PATH CALCULATION PROGRAM BASED ON CODE GENERATION TECHNIQUE

S. GOTO, T. OHTSUKI and T. YOSHIMURA
Central Research Laboratories, Nippon Electric Co., Ltd.
Kawasaki, Japan

ABSTRACT

A new computer implementation technique of shortest path algorithm is presented. By symbolic processing, a computer program generates another program which represents an optimal shortest path algorithm, in the sense that only non-trivial operations required for a given paticular network structure are excuted. The method presented here is powerful when a network of fixed sparseness structure must be solved repeatedly with different numerial values.

1. Introduction

The shortest path problem is one of the most fundamental problems in the area of network programming, and various algorithms have been proposed(1)~(3). The algorithm used here is the one which has an algebraic analogy with the well-known Crout elimination for solving linear algebraic equations. This paper presents a new implementation technique, in which a computer program generates, by symbolic processing, another program which represents an optimal shortest path algorithm of Crout elimination type, in the sense that only non-trivial operations required for a given particular network structure are executed. The implementation technique can be viewed as an application of "code generation technique" for solving a set of linear algebraic equations(4) to the shortest path problem, and is particularly powerful when a network of fixed structure must be solved repeatedly with different numerical values of edge-lengths. In example networks, the generated code was observed to run five- to eight-fold as fast as existing shortest path programs.

Unlike Potential method(1) or Dijkstra method(2), the shortest path algorithm of Crout elimination type enables us not only to apply "code generation technique" but also to estimate execution time in terms of network structure only (independent of numerical values of edge-lengths). Such a feature is suitable to those applications, e.g., automobile traffic control system, in which real-time compution of shortest paths is requested.

2. Preliminaries

For our purpose a digraph is a pair $G=(V,E)$, where V is a finite set of $n=|V|$ elements called vertices and E is a set of $m=|E|$ ordered vertex pairs called edges. For each edge $e=(u,v)\in E$, u and v are called its initial and terminal vertices, respectively.

A nonnegative weight $w(e)$, called length, is assigned for each edge $e\in E$. A finite sequence $P=\{v_1, v_2, \dots, v_\ell\}$ of distinct vertices is called a path from v_1 to v_ℓ if $(v_i, v_{i+1})\in E$ for $i=1, \dots, \ell-1$, and

$$W(P) = \sum_{i=1}^{\ell-1} W(e_i) \quad ; \quad e_i = (v_i, v_{i+1}) \tag{1}$$

is called the length of P. For an ordered pair (s,t), the set of paths from s to t is denoted by $\mathcal{P}(s,t)$. If

$$w(P) \leqq w(P') \tag{2}$$

for any $P' \in \mathcal{P}(s,t)$, then P and w(P) are called the shortest path and the distance from s to t, respectively.

Let $V = \{v_1, \ldots, v_n\}$ be the set of vertices of $G=(V,E)$, then the lengths of edges are represented by the incidence matrix $A = \{a_{ij}\}$, where

$$a_{ij} = \begin{cases} w(e_{ij}) \; ; & i \neq j \text{ and } e_{ij} = (v_i, v_j) \in E \\ \infty \; ; & i = j \text{ or } (v_i, v_j) \notin E \end{cases} \tag{3}$$

The implementation technique proposed here takes into account the sparseness of A, i.e., the property that the almost all elements of A are ∞. In our computer program, ∞ elements are never stored and operated.

Let x_{ij} be the distance from vertex v_i to v_j for each i, j=1,2, ... n. Then the $n \times n$ matrix $X = \{x_{ij}\}$ is called the distance matrix of the network, where diagonal elements x_{ii} ; i = 1,2,.. n, are considered to be 0 in a natural sense.

To represent the set of shortest paths from v_i to v_j ; i, j = 1,2..,n, we shall introduce the shortest path matrix $R = \{r_{ij}\}$ defined as follows. Let $P_{ij} = \{v_i, \ldots, v_k, v_j\}$ be the shortest path from v_i to v_j, then, if $r_{ij}=k$, it means that the vertex v_k is immediately precedent to the terminal vertex v_j in the sequence P_{ij}. In this case, it is clear that

$$x_{ij} = x_{ik} + x_{kj} = x_{ij} + a_{kj} \tag{4}$$

Thus the distance x_{ij} from v_i to v_j can be decomposed as the sum of the distance x_{ik} from v_i to v_k and the length a_{kj} of the edge (v_k, v_j). In this way, the sequence of vertices of each shortest path can be easily be found from the matrix R. For example if $r_{12}=7$, $r_{17}=3$ and $r_{13}=1$, it means that the stortest path P_{12} from v_1 to v_2 is given by

$$P_{12} = \{v_1, v_3, v_7, v_2\} \tag{5}$$

In summary, the shortest path problem is regarded as to evaluate some specified rows or columns of X and R for a given matrix A.

The wide variety of existing shortest path algorithms can be, basically, devided into two classes. In the first class, called "labelling method", Potential method(1) and Dijkstra method(2) are well-known. The algorithms of the other class are those which have an analogy with the elimination procedure for solving linear algebraic equations or for matrix inversion. Each elimination method, such as Gaussion elimination, Crout elimination and Gauss-Jordan elimination, yields the corresponding shortest path algorithm by taking the algebraic analogy between two problems into consideration. Particularly, the shortest path algorithm corresponding to Gauss-Jordan elimination is well-known by the name of Floyd method(6).

Table 1 summarizes the maximum number of basic operations which can be involved in each algorithm.

An important factor which affects the efficiency of shortest path algorithms is the number of starting points (denoted by q) specified in advance. As is seen from Table 1, the Dijkstra method is the most efficient if $q \ll n$(7). When q is close enough to n, on the other hand, all the elimination methods are more efficient than the Dijkstra method. Another advantage of the elimination methods, which is more important, is that their computational flow is independent of numerical values of edge-lengths. Unlike potential method or Dijkstra method, this favarable-property enables us to exploit "code generation technique" efficiently.

3. Crout elimination in the shortest path problem

Let x and y be non-negative numbers and define two basic operations as follows.

$$x \oplus y = \min\{x, y\} \tag{6}$$
$$x \otimes y = x + y \tag{7}$$

When x and/or y are ∞, these operations are interpretted in the natural way.

$$x \oplus \infty = x \tag{8}$$
$$x \otimes \infty = \infty \tag{9}$$

The basic operations involved in linear algebraic equations are addition/subtraction and multiplication. According to an algebraic consideration, it can be observed that multiplication and addition/subtraction in the matrix inversion correspond to addition and comparison in the shortest path problem, respectively. Table 2 shows the algebraic analogy between shortest path and matrix inversion problem.

The distance x_{ij}, from vertex v_i to vertex v_j is represented by

$$x_{ij} = \begin{cases} \min\limits_{k} \{ x_{ik} + a_{kj} \} & ; \ i \neq j \\ 0 & ; \ i = j \end{cases} \tag{10}$$

By using the above two operations, Eq. (10) can be written in matrix from(3):

$$X = A \otimes X \oplus I_n \tag{11}$$

where, In is the square matrix of order n with 0's in the diagonal position. It is considered to be the unit matrix in the shortest path problem.

Among these elimination methods listed in Table 1, Gauss-Jordan elimination is inferior to the others from the viewpoint of preserving sparsity(5). Comparing Gaussian elimination and Crout elimination, Crout elimination is slightly better, because comparision if n items simultaneously can be executed faster than the comparisions of two items (n-1) times. For this reason, the Crout elimination algorithm is implemented in our computer program.

According to an algebraic consideration(3), it is seen that Eq. (11) has an analogy with the equation for the matrix inversion. Taking this into account, we have:

$$X = \prod_{i=1}^{n-1} U^{(i)} \otimes \prod_{i=1}^{n-1} L^{(n-i+1)} \otimes I_n \tag{12}$$

where, $U^{(i)}$'s and $L^{(n-i+1)}$'s are of the form:

$$U^{(i)} = \begin{array}{c} \\ \\ i \\ \\ \\ \\ \end{array} \left(\begin{array}{cccccc} 0 & & & & & \\ & 0 & & \infty & & \\ & & 0 & x & x\ x \cdot \cdot & x \\ & & & \cdot & \infty & \\ & & & & \cdot & \\ & \infty & & & & 0 \end{array} \right) \tag{13}$$

$$L^{(n-i+1)} = \begin{array}{r} \\ \\ \\ \\ n-i+1 \\ \\ \end{array} \begin{bmatrix} 0 & & & & & \\ & 0 & & & \infty & \\ & & 0 & & & \\ & \infty & & \ddots & & \\ X & X & X & \cdots & X & 0 \\ & & \infty & & & 0 \end{bmatrix} \tag{14}$$

The expression (12) can be viewed as a shortest path version of the elimination form of inverse$^{(5)}$. Now the elements of $U^{(i)}$'s, $L^{(n-i+1)}$'s and specified rows (columns) of X can be evaluated by means of an obvious modification of triangular factorization, forward substitution and backward substitution procedures for solving linear algebraic equations. The shortest path matrix R can also be obtained by applying simple additional operations to the elimininati on process. It should be noted here that the computational flow of the algorithm is independent of numerical values, which makes code generation effective.

In the triangular factorization for obtaining the elimination form of inverse (12), the rows (columns) of incident matrix A are processed one by one. When A is sparse, the number of nontrivial operations, i.e., computation time required to obtain the specified rows (columns) of X and R, differs depending on pivoting order, the order in which the rows (columns) of A are processed. In this sense, it is an important problem to a priori find a good pivoting order from the topological structure of the given network. In determing a good pivoting order, we have to take the following two factors into account. First, the rows (columns) correspoinding to the specified starting points should be pivoted last, so that number of nontrivial operations in forward substitution is reduced. Secondly, the pivoting order should be determined, subject to the above constraint, in such a way that as few fill-ins (finite-valued elements) as possible are created in deriving (12).

Although, no efficient algorithm to determine the pure optimum pivoting order is known up to now, several heuristic algorithms which give statistically good pivoting orders have been proposed$^{(9)-(10)}$. Among them, the Markowitz's method$^{(10)}$ is used in our computer program.

4. Code Generation

The shortest path calculation by means of code generation, as illustrated in Fig. 1, consists of two stages, called set-up stage and execution stage. In the set-up stage, a computer program, by symbolic processing, 1) determines a good pivoting order according to the strategy described in the previous section and then 2) generate another program which represents the optimal reduced Crout algorithm, in the sense that only non-trivial operations required for a given network structure are excuted. In our computer implementation, the generated program is written in FORTRAN language and the source program is converted to a load module through a compiler and a linkage editor. The execution stage is, for given numerical values of edge lengths, to execute the load module, i.e., to calculate a set of shortest paths. Although the generated program cannot treat networks of different topological structure, it analizes the given network most efficiently. Therefore this method is extremely powerful when a fixed-structured network must be solved many times for different numerical values. It should be noted here that the set-up stage is processed once for a given network structure while the execution stage is processed many times.

As an example Fig. 3 gives the generated FORTRAN program for finding the shortest paths from vertex ① to all the other vertices in the network shown in Fig. 2. The program runs very fast, since it contains no DO-loop nor trivial operations (such as $a \oplus \infty = a$, $a \otimes \infty = \infty$). Furthermore it requires no address calculation, since the subscripts of arrays are of constants.

5. Computational Results

The following five shortest path programs run on IBM S/360 M195 were implemented to examine their efficiency. All programs are written in FORTRAN, except for a small part of P2 which is written in Assembler. Mesh-structured networks were provided as test data and randam values are assigned for edge-lengths.

P1: Code generation technique is applied to Crout elimination algorithm.

P2: Instead of generating code, an address table containing subscripts of array variables involved in non-trivial operations are generated. This program saves primary memory space by sacrificing execution speed compared with P1.

P3: Crout elimination algorithm is implemented without code generation.

P4: Dijkstra algorithm is implemented with a sophisticated list processing(11) in order to efficiently sort temporary-labeled vertices.

P5: Potential method is implemented.

Table 3 compares P1, P2 and P3 in terms of computation time to find the shortest paths from a specified vertex to all vertices, where a mesh-structured network of 64 vertices is taken as test data. P1 requires much computation time at the set-up stage. However it attains almost the ultimate limit of execution speed. P3 requires less time at the set-up stage, but it is less efficient than P1 at the excution stage. The computation time of P2 at the set-up stage and the execution stage is between that of P1 and P3.

Table 4 compares programs P1~P5 in terms of execution time to find all shortest paths. P3 is slightly slower than P4 or P5. However P1 and P2, due to the code generation or table generation, are much faster than P4 or P5.

Table 5 compares programs P1~P5 in terms of execution time to find shortest paths from q specified vertices to all vertices. P1 and P2 are not efficient if only one vertex is specified as starting point. However, these programs run faster than P4 or P5 when two or more starting points are specified.

6. References

(1) H. Frank and I. T. Frisch: Communication, transmission and transportation networks, Addison-Wesley, pp192~198(1971)

(2) E.W.Dijkstra: A note on two probelms in connection with graphs, Numerishe Mathematik, 1, P.269(1959)

(3) B.A. Carre: An algebra for network routing problems, J. Inst. Maths. Applics, Vol. 7, pp273~294(1971)

(4) F.G. Gustavson, W. Linger and R. Willoghby: Symbolic generation of optimal Crout algarithm for sparse systems of linear equations, JACM, Vol. 7, No.1 pp87~109(1970)

(5) R.P. Tewarson: Sparse Matrices, Academic Press, New York (1973)

(6) R.W. Floyd: Algorithm 97 - Shortest Path, C.ACM, 15, P.345(1962)

(7) S.E. Dreyfus: An appraisal of some shortest-path algorithms,

Oper. Res., 19, 3, p.395(1969-5)
(8) E.C. Ogbuobiri, W.F. Tinney and J.K. Walker: Sparsity directed decomposition for Gaussion elimination on matrices, IEEE. Trans., PAS-89, 1, P.141(1970-1)
(9) D.J. Rose: Triangulated graph and elimination process, J. Mathe. Anal. and Appli. pp597~609, 32(1970)
(10) Markowitz: The elimination of the inverse and its application to linear programming, Manage. Sci (1957-3)
(11) E.L.Johnson: On shortest paths and sorting, Proc, ACM, pp510~517 (1972-8)

	Comparison	Addition
Potential Method*	$n^3 q$	$n^3 q$
Dijkstra Method*	$3n^2 q$	$1/2n^2 q$
Gaussian Elimination**	$1/3n^3 + 1/2n^2 q + 1/6q^3$	$1/3n^3 + 1/2n^2 q + 1/6q^3$
Crout Elimination**	〃	〃
Gauss-Jordan Elimination**	$1/2n^3 + 1/2nq^2$	$1/2n^3 + 1/2nq^2$

Table 1. Number of operations in shortest path algorithms

n : number of vertices

q : number of starting points

*Labelling Method : depends on numerical values

**Elimination Method : independent of numerical values

Shortest Path	Matrix Inversion
$x \oplus y$	$x + y$
$x \oplus \infty = x$	$x + 0 = x$
$x \otimes y$	$x \times y$
$x \otimes \infty = \infty$	$x \times 0 = 0$
$x \otimes 0 = x$	$x \times 1 = x$
∞	0
0	1
$X = A \otimes X \oplus I_n$	$A X = I$

Table 2. Algebraic analogy between shortest path and matrix inversion problems

		P1(ms)	P2(ms)	P3(ms)
Set-up stage	Optimal Pivoting Order	896	896	896
	Triangular factorization	1051	33	-
	Forward substitutions	101	0.8	-
	Backward substitutions	384	3	-
	Compile and Link-Edit	31000	-	-
Execution stage	Triangular factorization	0.8	1.4	17
	Forward substitutions	0.06	0.09	0.4
	Backward substitutions	0.22	0.34	1.3

Table 3. Computation time to find shortest paths from a specified vertex to all vertices.

Number of vertices	P1(ms)	P2(ms)	P3(ms)	P4(ms)	P5(ms)
25	2	3	15	13	12
36	5	10	34	28	24
49	9	16	69	50	43
64	18	31	120	90	80
81	30	46	213	145	120
100	47	76	330	231	205

Table 4. Execution time to find all shortest paths

Number of starting points	P1(ms)	P2(ms)	P3(ms)	P4(ms)	P5(ms)
1	2.3	4.2	18.8	2.0	1.8
2	2.6	4.9	20.4	4.1	3.7
3	3.1	5.6	22	6.3	5.7
10	6.5	10	34	22	19
50	23	38	97	112	101
100	47	76	330	231	205

Table 5. Execution time to find shortest paths from q specified vertices to all vertices

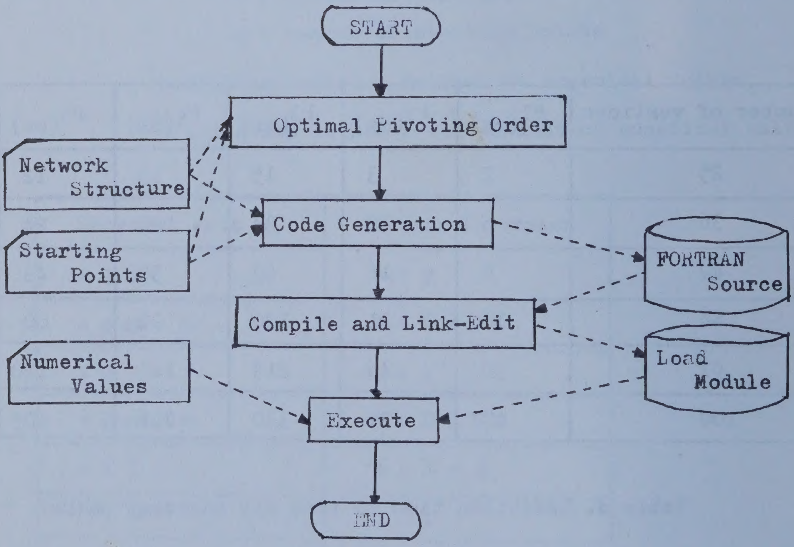

START
Optimal Pivoting Order
Network Structure
Code Generation
Starting Points
FORTRAN Source
Compile and Link-Edit
Numerical Values
Load Module
Execute
END

Fig.1. Code Generation Technique

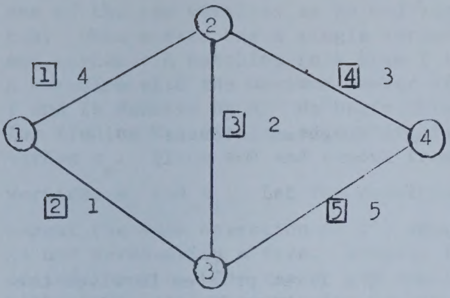


Fig.2. An example network

ⓘ : vertex

$\boxed{i}$: edge

i : length of an edge

```
C     TRIANGULAR FACTORIZATION
      VLU = A(3) + A(4)
      IF (VLU.GE.A(5)) GO TO 10
      A(5) = VLU
      R(5) = R(3)
C     INITIAL SET FOR SUBSTITUTIONS
   10 S(1) = 0
      X(1) = 0
      X(2) = 30000
      X(3) = 30000
      X(4) = 30000
C     FORWARD SUBSTITUTIONS
      X(2) = A(1)
      X(3) = A(2)
      S(2) = R(1)
      S(3) = R(2)
      VLU = A(3) + X(2)
      IF (VLU.GE.X(4)) GO TO 20
      X(4) = VLU
      S(4) = R(3)
   20 VLU = A(5) + X(3)
      IF(VLU.GE.X(4)) GO TO 30
      X(4) = VLU
      S(4) = R(5)
C     BACKWARD SUBSTITUTIONS
   30 VLU = A(4) + X(3)
      IF (VLU.GE.X(2)) GO TO 40
      X(2) = VLU
      S(2) = R(4)
   40 COTINUE
      RETURN
      END
```

Fig. 3 Generated Program

EMBEDDING AND EXISTENCE PROBLEMS IN TREES[1]

A. N. PATRINOS
International Business Machines
Greece

S. L. HAKIMI
Departments of Electrical Engineering & Computer Sciences
Northwestern University
Evanston, Illinois

ABSTRACT

Two kinds of problems are considered. The first problem involves the existence of certain types of subtrees in a tree. Specifically, O(p) algorithms are given to determine the maximum number of vertex-disjoint subtrees which are isomorphic to a given "star tree" or "line tree" of a tree. The other problem involves the existence of a tree associated with a given integer sequence. Applications are given.

1. Introduction

A graph g is said to be embedded in another graph G if G has a subgraph which is isomorphic to g. In general, to determine whether or not g is embedded in G is very difficult and it also leads to another even more general problem: how many "copies" of g are embedded in G? Or, more precisely, how many vertex-disjoint subgraphs of G are isomorphic to g? We note that if g consists of a single edge on a pair of vertices, then the above problem is identical with the celebrated matching problem in graph theory which has been solved [1-5]. If g is more complex then the problem is much more difficult. In Section 2, we will solve this problem when G is a tree (connected circuitless graph) and g is either a "star tree" or a "line tree".

Let T be a tree with p vertices and whose edges are $e_1, e_2, \ldots, e_{p-1}$. The removal of e_i from T, disconnect the tree into two parts with n_1^i and n_2^i vertices, with of course $n_1^i + n_2^i = p$. Let $a_i = \min(n_1^i, n_2^i)$. We associate with edge e_i the positive integer a_i. Let $A(T) = a_1 a_2 \ldots a_{p-1}$ be the sequences of integers associated with edges $e_1 e_2 \ldots e_{p-1}$. Let us now renumber the edges of T (if necessary) such that $a_i \leq a_{i+1}$ for i = 1,2,...,p-2. Therefore, with each tree T, we can associate a non-decreasing sequence of positive integers $A(T) = a_1 a_2 \ldots a_{p-1}$. The problem is: given a non-decreasing sequence of positive integers $\alpha_1 \alpha_2 \ldots \alpha_{p-1}$ is there a tree T such that $A(T) = \alpha_1 \alpha_2 \ldots \alpha_{p-1}$. This problem will be dealt with in Section 3. In Section 4, we give some problems and applications. The reader is expected to be familiar with standard terminology in graph theory [6].

2. Embedding or Generalized Matching Problems in Trees

An end vertex and an internal vertex of a tree T with p vertices

---

[1] This work is supported by the U.S. Air Force Office of Scientific Research, Systems Command, Grant AFOSR-71-2103 Amend D.

($p > 2$) is a vertex of degree 1 and a vertex of a degree greater than one, respectively. If a tree has two vertices, then we arbitrarily designate one of the two vertices as an end vertex and the other as an internal vertex. When a tree has a single vertex, then that vertex is designated as an end vertex. A matching in a tree T is a set of vertex-disjoint edges in T. A matching with the maximum number of edges is called a maximum matching of T and is denoted by M. We begin this section by giving a simple algorithm for finding M. Pick an edge $e=(v_e,v_i)$ adjacent to (or incident at) an end-vertex v_e. Place $e \in M$ and remove from T all edges incident at v_i, and both vertices v_e and v_i. Let the resulting graph be T'. If T' has any edges, repeat the same operation on T', and stop otherwise. It should be noted T' is not necessarily a tree. However T' is circuitless and such graphs are called forests. The reader can easily verify the correctness of the algorithm and that its complexity is bounded by O(p) steps. Now we turn our attention to more complex matching type problems on trees. Before that we introduce a labeling algorithm for trees.

2.1 A Labeling Algorithm for Trees

By removing an end-vertex we imply a deletion of the vertex together with its incident edge. We now describe a systematic way of visiting vertices in a given tree T. Given that Q is an originally empty (first in first out) FIFO store we shall adopt the following order:

Step i: Visit the end-vertices of T(i) in an arbitrary order ($i \geq 1$ and $T(1) = T$). Each end-vertex visited is assigned the label $i - 1$ and added to Q. Remove the end-vertices of T(i). Let $T(i+1)$ be the resulting tree. If $T(i+1)$ is empty, terminate the algorithm.

The algorithm needs $\left\lceil \frac{\delta + 3}{2} \right\rceil$ steps to terminate, where δ is the diameter of the tree. Q will contain the vertices in the order by which they should be traversed in the applications described below. The label of a vertex v is denoted by $\ell(v)$.

We state the following theorem whose proof is left to the reader:

Theorem 1: Let the label associated with vertex v in tree T by the preceding algorithm be $\ell(v)$, then

a) For every vertex v in T there exists at most one other adjacent vertex, say v', for which $\ell(v) < \ell(v')$.

b) For every vertex v with $\ell(v) > 0$ there exists an adjacent vertex u with $\ell(u) = \ell(v) - 1$.

c) There exists exactly one vertex with the maximum label.

d) Two vertices with equal labels cannot be adjacent.

2.2 Star Tree Matching

A star tree, denoted by S_k, is the tree with k edges and a vertex of degree k. We shall determine the maximum number of vertex-disjoint trees isomorphic to S_k embedded in a given tree T. Since the case $k = 1$ has been considered we shall assume $k > 1$.

A matching, here, will be defined as a set of vertex-disjoint subtrees isomorphic to S_k which are embedded in T. Each element of a matching will be defined by the set of its constituent edges. A maximum matching $S_k(T)$ will be defined as a matching of maximum cardinality. A maximum matching may not be unique. Moreover if no vertex in T has degree greater than

$k-1$ then, clearly, $|S_k(T)| = 0$. By removing an edge we imply a simple erasure of the connection between the two vertices at which the edge is incident. By removing a vertex we mean removing that vertex and all edges incident to it.

Let us assume that the labeling algorithm has been applied to T and let v be a vertex in T with $\ell(v) = 1$. Such a vertex always exists $(p > 1)$. Let $\{v_1, v_2, \ldots, v_r\}$ be the set of all vertices adjacent to v with $\ell(v_i) < \ell(v)$, $i = 1,2,\ldots,r$.

Theorem 2: (a) If there exists a vertex v' adjacent to v with $\ell(v') > \ell(v)$ as in Fig. 1, then we have three possibilities.

a1) $r < k-1$. Then none of the edges $(v,v_1), (v,v_2) \ldots (v,v_r)$ can be part of any member of $S_k(T)$ and thus they could be removed from T.

a2) $r = k-1$. Then there exists a maximum matching $S_k(T)$ such that $\{(v,v'), (v,v_1), (v,v_2), \ldots, (v,v_k)\} \in S_k(T)$

a3) $r > k-1$. Then there exists a maximum matching $S_k(T)$ such that $\{(v,v_1), (v,v_2), \ldots, (v,v_k)\} \in S_k(T)$

(b) If there does not exist a vertex v' such that $\ell(v') > \ell(v)$, then we have two possibilities:

b1) $r \leq k-1$. Then none of the edges $(v,v_1), (v,v_2), \ldots, (v,v_r)$ can be a member of $S_k(T)$ and thus they could be removed from T

b2) $r > k-1$. Then there exists a maximum matching $S_k(T)$ such that $\{(v,v_1), (v,v_2), \ldots, (v,v_k)\} \in S_k(T)$.

The proof of Theorem 2 although a bit long is self-explanatory and therefore is omitted.

Theorem 2 suggests an algorithm for finding $S_k(T)$ as follows. Having labeled the tree with the "tree labeling algorithm", apply the theorem to all vertices of label 1, then repeat for all vertices of label 2, and so forth. If at any step a member of $S_k(T)$ is found, remove from T that member of $S_k(T)$ and all vertices associated with it. Continue in the next step with resulting modified tree. It should be clear that the number of computation is again bounded by $O(p)$.

2.3 Line Tree Matching

The algorithm here finds the maximum number of vertex-disjoint trees isomorphic to a line tree which are embedded in a given tree T with p vertices. A line tree, denoted by L_k, is a tree with k edges and exactly two end-vertices $(k > 1)$. A matching, here, will be defined as a set of vertex-disjoint trees isomorphic to L_k embedded in T. A maximum matching, denoted by $L_k(T)$, will be defined as a matching of maximum cardinality. An element of $L_k(T)$ will be defined by the set of its constituent edges. If k is greater than the diameter δ of T then $|L_k(T)| = 0$. We shall take $k > 2$.

Let us assume that the labeling algorithm has been applied to T. Consider a vertex v in T. Let v' be the vertex adjacent to v (if one exists) with $\ell(v') > \ell(v)$. We define T(v) as that component of $T - (v,v')$ containing v, and p(v) as the path of maximal length in T(v) one of whose end-vertices is v where the length of a path p is denoted by $|p|$ and is equal to the number of edges in p. If v' does not exist then $T(v) = T$.

Let $v_1, v_2, \ldots v_r$ be the vertices adjacent to v with $\ell(v_i) < \ell(v)$, $i = 1,2,\ldots,r$, and assume the following:

a) $|L_k(T(v_i))| = 0$, $i = 1,2,\ldots,r$,

b) $|p(v_1)| \geq |p(v_2)| \geq \ldots \geq |p(v_r)|$.

Let $p(v_1) = \{(v_1, v_{1_1}), (v_{1_1}, v_{1_2}), \ldots, (v_{1_{s-1}}, v_{1_s})\}$ and

$p(v_2) = \{(v_2, v_{2_1}), (v_{2_1}, v_{2_2}), \ldots, (v_{2_{\ell-1}}, v_{2_\ell})\}$ (See Figure 2).

Theorem 3: Based on the above terminology, we have the following three cases.

a) $|p(v_1)| + 1 = k$. Then there exists a maximum matching $L_k(T)$ such that $\{(v,v_1), (v_1, v_{1_1}), \ldots, (v_{1_{s-1}}, v_{1_s})\} \in L_k(T)$.

b) $|p(v_1)| + |p(v_2)| + 2 \geq k$. Then there exists a maximum matching $L_k(T)$ such that $\{(v,v_1), (v_1, v_{1_1}), \ldots, (v_{1_{s-1}}, v_{1_s}), (v, v_2), (v_2, v_{2_1}), \ldots, (v_{2_{q-1}}, v_{2_q})\} \in L_k(T)$ where $q = k - s - 2$.

c) None of the above two. Then there exists no subtree of T(v) isomorphic to L_k. Clearly, $|p(v)| = |p(v_1)| + 1$.

The proof is again self-explanatory and thus omitted.

Theorem 3 suggests an algorithm for finding $L_k(T)$. We proceed by visiting vertices. The order of visiting vertices in T, as vertex v above, is the order by which vertices are removed from Q. Whenever a member of $L_k(T)$ is found, then the theorem is applied to a new tree which after the first step is T(v'). Vertices with labels less than $\ell(v)$ have already been visited and the values $|p(v_1)|$, $|p(v_2)|, \ldots, |p(v_r)|$ have been calculated. A repetition of the above process results in a maximum matching $L_k(T)$. The complexity of the algorithm is bounded by O(p).

3. Integer Sequences and Trees

We begin with the following result:

Theorem 4: If $A(T) = (a_1, a_2, \ldots, a_{p-1})$ is the non-decreasing sequence of positive integers associated with a tree T with p vertices then:

(1) $a_i \leq (i+1)/2$,

(2) if $a_i = a_{i+1} =, \ldots, = a_{i+(r-1)} = \alpha$, $(r > 1)$, then $r \cdot \alpha \leq i + r - 1$, and

(3) if in (2) we have $r \cdot \alpha = i + r - 1$ then $a_{i+r} = m \cdot \alpha + 1$, where m is a positive integer with $1 \leq m \leq r/2$.

Proof. (1) Let T be the tree with p vertices whose associated sequence is A(T). It can easily be seen that, by definition of the centroid of a tree [6], at least one of the two vertices connected by the edge e_{p-1}, whose label is a_{p-1}, is a centroid of T. By definition, $a_{p-1} = \min\{n_1^{p-1}, n_2^{p-1}\}$, where $n_1^{p-1} + n_2^{p-1} = p$. Therefore $a_{p-1} \leq p/2$.

Consider now the following operation: Remove the edge e_{p-1} and coalesce the two vertices it connects in T. The resulting tree T' has p-1 vertices and the positive integers (labels) associated with its edges are as before, because of the properties of the centroid of a tree. Therefore the sequence associated with T' is $A(T') = (a_1, a_2, \ldots, a_{p-2})$. By similar reasoning we have $a_{p-2} \leq (p-1)/2$. Repeating this procedure we conclude that $a_i \leq (i+1)/2$, $i = 1,2,\ldots,p-1$. This proves Condition (1). The argument used above also leads to the following observation: If $A(T^p) = (a_1,a_2,\ldots, a_{p-1})$ is the integer sequence associated with T^p then there exists a tree T^{i+1} for which $A(T^{i+1}) = (a_1,a_2,\ldots,a_i)$, $i = 1,2,\ldots,p-1$. (2) Let T^{i+r} be the tree satisfying the sequence $A(T^{i+r}) = (a_1,a_2,\ldots, a_i, a_{i+1},\ldots, a_{i+r-1})$. Since the members of $A(T^{i+r})$ are in non-decreasing order, and $a_i = a_{i+1} = \ldots = a_{i+r-1} = \alpha$, all the edges of T^{i+r} whose associated labels equal α are incident at the centroid of T^{i+r}. The tree T^{i+r} must have at least $r\cdot\alpha+1$ vertices (unity accounts for the centroid) because r edges have label equal to α. Therefore $i+r \geq r\cdot\alpha+1$. (3) If in (2) we have $r\cdot\alpha+1 = i+r$, then, in the tree T^{i+r} only the edges with label equal to α are incident at the centroid. Therefore, in the tree T^{i+r+1} satisfying the sequence $A(T^{i+r+1}) = (a_1,a_2,\ldots,a_i, a_{i+1},\ldots, a_{i+r})$, we must have $a_{i+r} = m\cdot\alpha+1$, where m is a positive integer. This, however, implies that $1 \leq m \leq r/2$; because, by Condition (1), $a_{i+r} \leq (i+r+1)/2$.

The three conditions given in the theorem are necessary but not sufficient. This is illustrated by the sequence (1, 1, 1, 2, 2, 2, 3, 3, 5, 5). This sequence satisfies Conditions (1), (2) and (3) but no tree exists which can be associated with this sequence.

Consider the following method of constructing a tree with p vertices (if one exists) which satisfies a non-decreasing sequence of positive integers $(a_1,a_2,\ldots, a_{p-1})$. We assume that the three conditions of the theorem are satisfied. To avoid unnecessary complications in notation we only give here an outline of the algorithm. Let v be a fixed vertex which is initially isolated. We shall build the tree, step by step, maintaining v as a centroid of all the trees constructed in the process. We shall use B_k to denote the sequence of labels associated with all the edges incident at v at the start of Step k. Let $a_1 = a_2, \ldots, = a_m = 1$ and $a_{m+1} > 1$.

Step 0.

Construct a tree as illustrated on Fig. 3. We now have $B_1 = (a_1,a_2,\ldots,a_m)$. If $m = p-1$, the algorithm terminates successfully.

Step k.

Let $B_k = (b_1,b_2,\ldots,b_s)$.

(a) Find, if possible, a subsequence $(b_{i_1},b_{i_2},\ldots, b_{i_r})$ of B_k such that $\sum_{j=1}^{r} b_{i_j} = a_{m+k}-1$, then construct a tree as shown on Fig. 4. If $m+k = p-1$, then the algorithm terminates successfully. If $m+k \neq p-1$, then set $k = k+1$ and perform Step k.

(b) Otherwise, look for the largest integer q ($q < k$) for which a different choice of a subsequence of B_q exists. If $q > 1$ then repeat, continuing from Step q, on this alternate choice. If $q = 1$, then the algorithm terminates unsuccessfully (i.e., no tree exists to satisfy the sequence).

The reader can verify that a depth-first search is used to arrive at a tree satisfying the sequence. It can also be seen that this algorithm is not polynomially bounded on p, making its implementation too complex on a time scale. We believe, however, that the problem cannot be solved efficiently. Note that at the start of any step in the above procedure, the sum of the labels of all edges incident at v is equal to the number of edges in the tree being considered at the start of the step.

The problem presented above has an interesting representation in zero-one programming. If $A = (a_1, a_2, \ldots, a_{p-1})$ is the given non-decreasing sequence of positive integers, let $\underline{v} = (a_1, a_2, \ldots, a_{p-1})$ and $\underline{u} = (1, 2, \ldots, p-1)$ be row vectors. Then, there exists a tree T with p vertices satisfying A if and only if there exists a binary matrix $M = \left[m_{ij} \right]$ of order p-1 such that $M \cdot \underline{v}^t = \underline{u}^t$, where

$$m_{ii} = 1,\ 1 = 1,2,\ldots,p-1$$

$$m_{ij} = 0,\ j > i, \text{ and}$$

$$m_{ij} \geq m_{(i+1)j},\ i \geq j$$

(Symbol "." denotes matrix multiplication and "t" denotes the transpose).

4. Conclusions and Applications

The problems discussed in Section 2 being generalizations of the matching problem have applications to "packing type" problems as well as in pattern recognition. The problem presented in Section 3 seems somewhat extraordinary; therefore, a few words about one of its potential applications is in order.

The problem in Section 3 is a rather special case of a knapsack-type problem. Consider a non-increasing sequence $A = (a_1, a_2, \ldots, a_{p-1})$ of positive real numbers. We construct a tree with p vertices to be associated with the sequence A as follows. Fix a vertex v and call it a 'root'.

Step 1.

Attach an edge with label a_1 to v as shown on Fig. 5.

Step k.

Let v_i be an arbitrary vertex, other than the root, on the tree resulting from Step k-1, and let $a_{i_1}, a_{i_2}, \ldots, a_{i_s}$ $(1 \leq i_j \leq k,\ j = 1,2,\ldots,s)$ be the labels of the edges incident at v_i. Without loss of generality, let a_{i_1} be the label of the edge incident at v_i closest to the root v. Compute $w(v_i) = a_{i_1} - \sum_{2}^{s} a_{i_j}$. Now attach an edge with label a_k either to a vertex v_k in the tree, for which $w(v_k) > a_k$, as shown in Fig. 6, or to the root.

The objective of the problem however, is to construct a tree in which the sum of the labels of the edges incident at the root v is the minimum possible. With this objective in mind, it can be seen that if a tree, of the previous section, exists which satisfies a non-decreasing sequence of positive integers obtained from A by reversing their order, it is an optimum solution to the problem of this section.

An application of this problem is the case were the members of the given sequence A are containers with volumes $a_1 a_2, \ldots, a_{p-1}$. If we remove the geometrical restrictions, we say that a container of volume a_i 'fits' into a container of volume a_j if and only if $a_i < a_j$. The problem would then be to find a container of minimum volume into which the members $a_1, a_2, \ldots, a_{p-1}$ can fit.

REFERENCES

1. Berge, C., "Two Theorems in Graph Theory", Proc. Nat. Acad. Sci., 43, (1957), 842-844.
2. Edmonds, J., "Paths, Trees and Flowers", Canad. J. Math., 17, (1965), 449-467.
3. Ford, L. R. and Fulkerson, D. R., Flows in Networks, Princeton University Press, Princeton, 1962.
4. Hopcroft, J. E. and Karp, R. M., "A $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs", SIAM J. Comp., 2, (1973), 225-231.
5. Berge, C., Graphs and Hypergraphs, North-Holland/American Elsevier, 1973.
6. Harary, F., Graph Theory, Addison-Wesley, Reading, Mass., 1969.

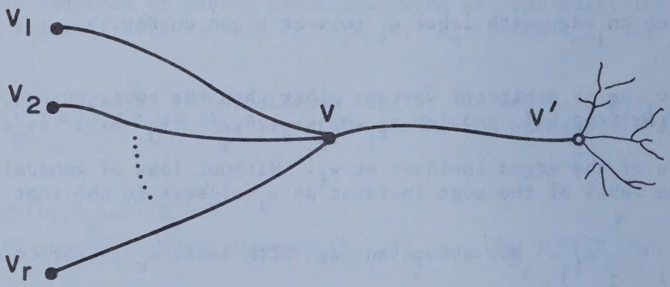


Figure 1

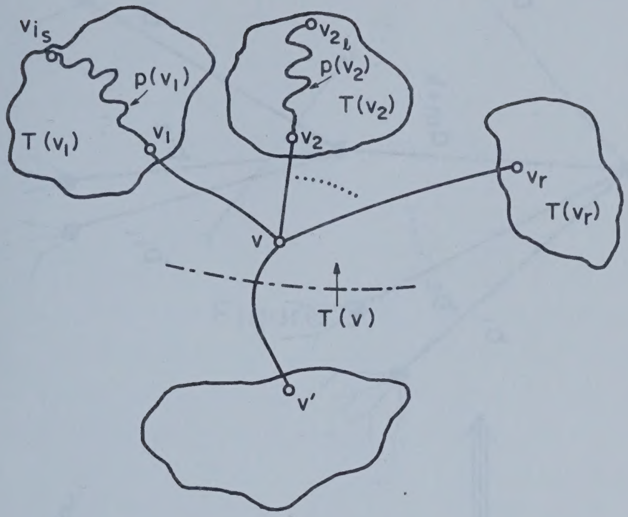

v_{i_s}
$p(v_1)$
$T(v_1)$
v_1
v_{2_ℓ}
$p(v_2)$
$T(v_2)$
v_2
v_r
$T(v_r)$
v
$T(v)$
v'

Figure 2

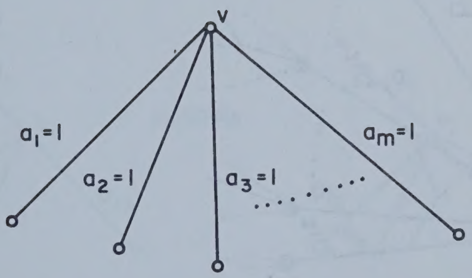

v
$a_1=1$
$a_2=1$
$a_3=1$
$a_m=1$

Figure 3

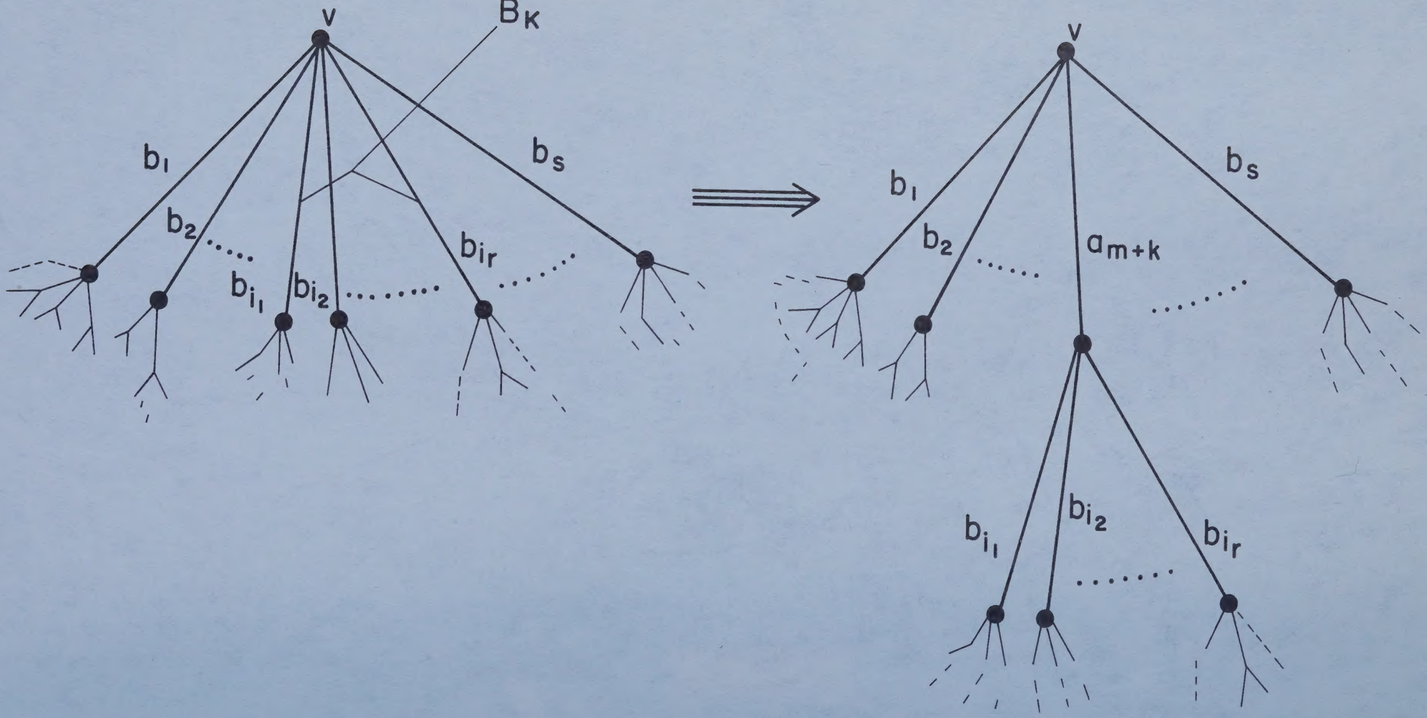

v
B_K
b_1
b_2
b_{i_1}
b_{i_2}
b_{i_r}
b_s
v
b_1
b_2
a_{m+k}
b_s
b_{i_1}
b_{i_2}
b_{i_r}

Figure 4

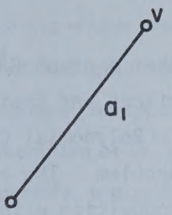


Figure 5

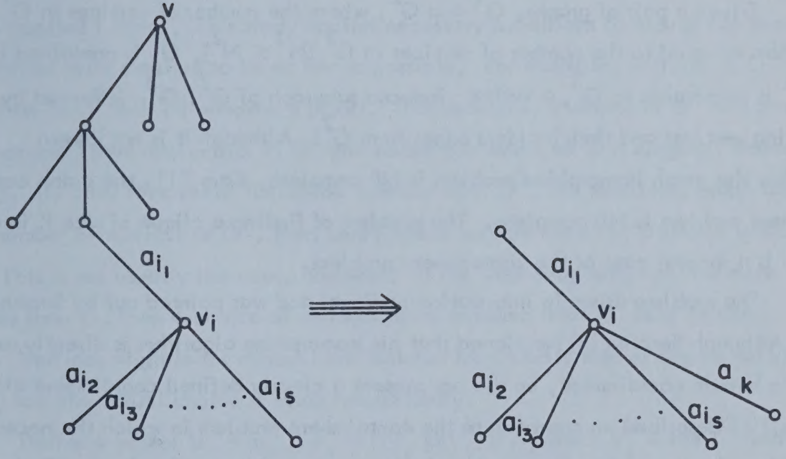

v
a_{i_1}
v_i
a_{i_2}
a_{i_3}
a_{i_s}
a_{i_1}
v_i
a_k
a_{i_2}
a_{i_3}
a_{i_s}

Figure 6

A BACKTRACKING ALGORITHM FOR TESTING VERTEX-INDUCED SUBGRAPHS

James D. Johannes
The University of Alabama
In Huntsville
Huntsville, AL 35807

Douglas C. Schmidt
Vanderbilt University
Nashville, TN 37235

Larry E. Druffel
USAF Academy
Colorado Springs, CO 80840

ABSTRACT

The problem of testing whether a graph G^1 can be made isomorphic to another graph G^2 by deleting vertices and their incident edges from G^2 is considered. Since the problem is Polynomial Complete, a backtracking algorithm is employed to solve the problem. The algorithm "proposes" candidate subgraphs of G^2 and tests the proposition with an isomorphism tester. Necessary conditions for containment are used to reduce the number of isomorphism tests as well as guide the search for candidates. In spite of the large upper bound on the number of graphs to be tested, the algorithm performs well.

THE PROBLEM

Given a pair of graphs, G^1 and G^2, where the number of vertices in G^1 is less than or equal to the number of vertices in G^2 ($N^1 \leq N^2$), G^1 is contained in G^2 if G^1 is isomorphic to G^3, a vertex induced subgraph of G^2. G^3 is formed by deleting vertices and their incident edges from G^2. Although it is not known whether the graph isomorphism problem is NP complete, Karp [1], the graph containment problem is NP complete. The problem of finding a clique of size K in a graph is a special case of the containment problem.

The problem arises in information retrieval and was pointed out by Sussenguth [2]. Although Berztiss [3] mentioned that his isomorphism algorithm is directly extendable to handle containment, he did not present a clearly defined containment algorithm. Corneil [4] described an approach to the containment problem in which the necessary conditions for isomorphism were relaxed. However, neither a clearly specified algorithm for containment nor run times have appeared in the literature.

This paper describes in detail an algorithm of the type Corneil discusses and presents computational results.

THE CONTAINMENT ALGORITHM

The Containment Algorithm reduces the size of G^2 by removing vertices and their incident edges to induce a subgraph G^3 which contains the same number of vertices and edges as G^1. The reduced graph G^3 will then be tested for isomorphism with G^1. While there is no algorithm known to be efficient for the isomorphism problem, there are a number of isomorphism algorithms which perform well in most cases [5-9]. Further, Druffel [10] has developed an algorithm which predicts the difficulty of testing a given graph for isomorphism and found that most graphs are amenable to graph isomorphism testing. Using this prediction technique, an upper bound on the time the isomorphism algorithm will take to test the isomorphism of G^1 with any other graph is established. This upper bound can be used to identify, a priori, some cases which will pose problems for the containment algorithm.

To gain an intuitive appreciation for the complexity of the containment problem, realize that the number of vertices which must be removed from G^2 is $N^2 - N^1$. There are $T = C^{N^2}_{N^1}$ graphs of size N^1 which may be induced from G^2. Since some of these graphs may not have the correct number of edges, $(M^3 \neq M^1)$, it may not be necessary to apply the isomorphism algorithm T times. However, there is no guarantee that it will not be applied T times. This study applies necessary conditions to reduce the number of isomorphism tests required to be as few as possible. For example, vertices in G^2 which cannot be in G^3 will be removed a priori. (For example, a vertex in G^2 may have lower degree, than any vertex in G^1 and hence can never be in a subgraph isomorphic to G^1). If, after removal of ineligible vertices from G^3, the remaining graph has the same number of vertices as G^1, then containment may be tested by a simple isomorphism test. This is not usually the case, however. If the remaining subgraph has more vertices than G^1, then a choice of vertices to be removed from G^2 must be made.

Vertices eligible for removal are isolated by a simple rule of degree satisfiability and the actual choice is made heuristically.

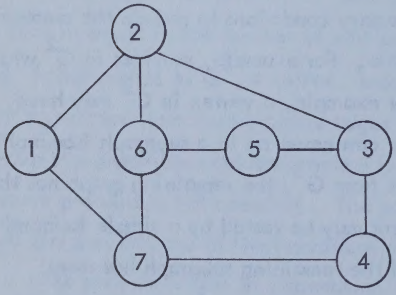
Define a vector $U^k = \{u^k_i\}$, $u^k_i = |\{V^k_j \mid d(V^k_j) > i\}|$ where V^k_j is the j^th^ vertex in G^k and $d(V^k_j)$ is its degree. If G^1 is to be isomorphic to G^2 then only vertices V^3_r $\ni U^3_{d(V^3_r)} > U^1_{d(V^3_r)}$, may be removed from G^2 to form G^3. This is the necessary condition which is used to identify candidates for removal from G^2. Such vertices will be termed degree satisfiable.

A graph containment algorithm based on degree satisfiability is outlined in Appendix 1. The algorithm backtracks making arbitrary vertex deletions (heuristically guided) followed by vertex forced deletions.

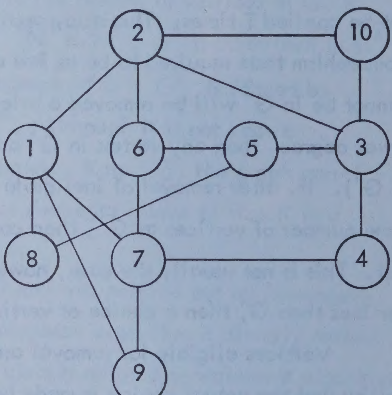
As stated in Appendix 1, the algorithm makes an arbitrary selection rule in Step 4. The choice rule employed in this study was to select an available candidate, V, from the set of vertices, $\{V|d(V) \leq K\}$, such that K is the minimum number where $U_k^3 > U_k^1$. This has the effect of imposing a strict limit on the choices at all levels.

EXAMPLE

Consider G^1 and G^2 in Figure 1, and their adjacency matrices in Figure 2. There is a vertex induced subgraph of G^1 in G^2 (vertices 1-7 in G^1 mapped to vertices 1-7 in G^2). Appendix 2 is a summary print-out of the choices made for containment testing of G^1 in G^2. In order to find the containment, the isomorphism algorithm was evoked once which is a substantial improvement over the 330 times required by brute force. The algorithm was run against 80 pairs of random graphs. Run times are summarized in Figure 3. While this is not overwhelmingly fast, the times are on the order of times for several early isomorphism testers.



Graph 1. An induced subgraph of Graph 2.



Graph 2. A supergraph of Graph 1.

Figure 1

ADJACENCY MATRIX FOR G^1

$$
\begin{array}{c|ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
3 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
4 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
5 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
6 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
7 & 1 & 0 & 0 & 1 & 0 & 1 & 0
\end{array}
$$

ADJACENCY MATRIX FOR G^2

$$
\begin{array}{c|cccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
3 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
4 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
5 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
6 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
7 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
8 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
9 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
10 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Figure 2

Average Containment Processing Time

Versus Graph Size

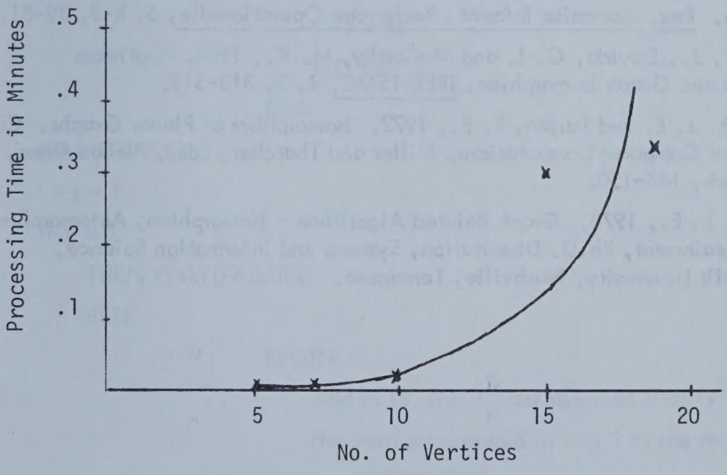


Figure 3

SUMMARY

The algorithm is performing in the same range as early Isomorphism Algorithms. It is felt that stronger heuristics would provide better backtracking performance.

The current algorithm performs very well when the subgraph and graph are of similiar size. An alternate approach is being investigated for small subgraphs which are less than half the size of the graph. The approach is to grow the subgraph to the graph as opposed to shrinking the graph to the subgraph.

REFERENCES

1. Karp, R. M., 1972. Reducibility Among Combinatorial Problems. Technical Rep. No. 3, Comp. Sc. Dept., Univ. of Cal. - Berkeley.

2. Sussenguth, E. H., Jr., 1964. A graph Theoretic Algorithm For Matching Chemical Structures, J. Chem. Doc., 5, 1, 36-43.

3. Berztiss, A. T., 1973. A Backtrack Procedure For Isomorphism of Directed Graphs, J. ACM, 20, 3, 365-377.

4. Corneil, D. G., 1968. Graph Isomorphism, Ph.D. Dissertation, Dept. of Comp., Sc., Univ. of Toronto, Toronto, Ontario, Canada.

5. Corneil, D. G., and Gotlieb, C. G., 1970. An Efficient Algorithm For Graph Isomorphism, J. ACM, 17, 1, 51-64.

6. Salton, G. and Sussenguth, E. H., Jr., 1964. Some Flexible Information Retrieval Systems Using Structure Matching Procedures, A.F.I.P.S., S.J.C.C.

7. Saucier, G., 1971. Un Algorithme Efficace Recherchant l'isomorphisme de 2 Graphes. Rev. Francaise Infomat. Recherche Operationelle, 5, R-3, 39-51.

8. Shah, Y. J., Davida, G. I. and McCarthy, M. K., 1974. Optimum Features and Graph Isomorphism, IEEE-TSMC, 4, 3, 313-319.

9. Hopcroft, J. E. and Tarjan, R. E., 1972. Isomorphism of Planar Graphs. Complexity of Computer Computations, Miller and Thatcher, Eds., Plenum Press, New York, 143-150.

10. Druffel, L. E., 1975. Graph Related Algorithms - Isomorphism, Automorphism and Containment, Ph.D. Dissertation, Systems and Information Science, Vanderbilt University, Nashville, Tennessee.

APPENDIX 1

CONTAINMENT ALGORITHM

```
BEGIN
  0  (Initialize) $G^3 \leftarrow G^2$; $t \leftarrow 0$; $a \leftarrow \mathrm{MIN}\ d(V_i^1)$
  1. (Extract Impossible Vertices)
     WHILE $(\exists (d(V_r^3) < a))$ DO
     Delete such a vertex from $G^3$
  2. (Test the possibility of a subgraph)
     IF $(M^3 < M^1 \vee N^3 < N^1 \vee (N^3 = N^1 \wedge M^3 \neq M^1) \vee (U_k^3 < U_k^1)$ THEN GO TO 5
  3. (If the proper number of vertices remain test for isomorphism)
     IF $N^3 = N^1$ THEN
        IF $G^1$ isomorphic to $G^3$ then HALT(SUCCESS) ELSE GO TO 5
  4. (Select a candidate vertex)
     IF there exists a vertex $V_r^3$ which has not been selected at a level $s \leq t$ since
     the last visitation to level $s - 1$ such that $U^3_{d(V_r^3)} > U^1_{d(V_r^3)}$ and $\nexists$ a $K < d(V_r^3)$
     where $U_k^3 = U_k^1$
                       THEN
                          BEGIN
                              Remove such a vertex,
                              $t \leftarrow t + 1$;
                              GO TO 1;
                          END
  5. (BACKTRACK)
     $t \leftarrow t - 1$
     IF $t < 0$
        THEN HALT(FAILURE)
        ELSE
                          BEGIN
                              Add to $G^3$ the $V_r^3$ last removed from $G^3$ in step 4 and
                              the vertices removed in step 1 as the result of the
                              removal of $V_r^3$;
                              GO TO 4;
                          END
END
```

APPENDIX 2

TRACE OF THE ALGORITHM FOR THE EXAMPLE

```
LEVEL 0
      D(V1)  2 3 3 2 2 3 3
      U1     7 7 4
      D(V3)  4 4 4 3 3 3 4 2 3 2
      U3     10 10 8 4
LEVEL 1
      K = 2  R = 10  N3 = 9  E3 = 14
      D(V3)  4 3 3 3 3 3 4 2 3 0
      U3     9 9 8 2
LEVEL 2
      K = 2  R = 8  N3 = 8  E3 = 12
      D(V3)  3 3 2 3 0 2 4 0 3 0
      U3     8 8 7 1
LEVEL 3
      K = 2  R = 5  N3 = 7  E3 = 10
      D(V3)  3 3 2 3 0 2 4 0 3 0
      U3     7 7 5 1
LAST V3(R) RETURNED
LEVEL 2
      K = 3  R = 9  N3 = 7  E3 = 9
      D(V3)  2 3 3 2 2 3 3 0 0 0
      U3     7 7 4 0
*     ISOMORPHIC TEST
      "GRAPHS ARE ISOMORPHIC"
      TIME = 12
```

AMPLITUDE FILTERING USING A TIME VARIABLE TRANSFORMATION*

ELLIOT SINGER AND ALAN V. OPPENHEIM
Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
Cambridge, Massachusetts 02139

ABSTRACT

This paper presents an analysis and clarification of a technique known as E-filtering which has been proposed by Moore and Parker for performing amplitude filtering on a signal. An alternative E-filter implementation is presented which consists of independent invertible subsystems. Although the alternative system is shown to be homomorphic, simulations indicate that it introduces severe time distortions at its output.

INTRODUCTION

In a recent paper[1] Moore and Parker described a filtering scheme for separating signals by their amplitude characteristics rather than by their frequency spectra. The filter, called an E-filter, distorts the time variable of the signal so as to map segments with different amplitude characteristics into different frequency bands even if their original spectra overlap. One application for such a filter is the removal of additive noise from a broadband signal without smoothing sharp edges in the signal. Alternatively, the system could be used as a selective peak filter, passing the high-amplitude signal peaks while attenuating the low-amplitude peaks. We report here on a project that analyzed and evaluated the E-filter. One aspect has been the determination of an alternative implementation of the E-filter which lends itself more easily to available analytic techniques. Although the alternative scheme is shown to be less practical than the original, its derivation clarifies some of the issues surrounding E-filters.

E-FILTERS

A representation of the E-filter proposed by Moore and Parker is shown in Fig. 1. The system A_o performs a transformation that "replots" the input signal s(t) against a new "time" variable e, where e is the arc length accumulated by s(t) up to time t. Thus $A_o\{s(t)\}$ is defined

---

*This work was supported in part by the Advanced Research Projects Agency monitored by ONR under Contract N00014-75-C-0951 and in part by the National Science Foundation under Grant ENG71-02319-A02.

by the relation

$$f(e)\Big|_{e=\theta(t)} = s(t), \tag{1a}$$

where

$$\theta(t) = \int_0^t \sqrt{1 + \dot{s}^2(\tau)}\, d\tau \tag{1b}$$

and $\theta(0) = 0$.

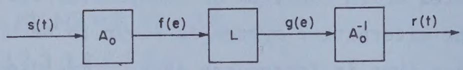


Fig. 1: E-filter proposed by Moore and Parker[1]. Note that g(e) is the equivalent of Moore and Parker's label f*(e).

The transformation performed by A_o is illustrated in Fig. 2. The system L with input f(e) and output g(e) is modeled as an ideal lowpass filter. The signal g(e) is the input to the system A_o^{-1} which produces the final output r(t). A_o^{-1} corresponds to the implementation used by Moore and Parker to invert the input transformation A_o; that is, $A_o^{-1}\{g(e)\}$ is defined by the relation

$$r[\theta^{-1}(e)] = g(e) \tag{2a}$$

or

$$r(t) = g(e)\Big|_{e=\theta(t)}. \tag{2b}$$

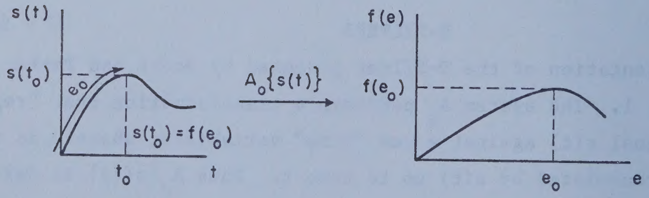


Fig. 2: Transformation performed by A_o.

As shown in Fig. 3 the mapping used by A_o to transform from t to e is now reversed so as to produce a value t for every corresponding value of e. Thus the output r(t) of the E-filter is the function that when plotted against the arc length of s(t) produces g(e). It is useful to interpret the transformations performed by the systems A_o and A_o^{-1} as instantaneous time warpings of their input signals. Thus $f(e) = f[\theta(t)]$ corresponds to a time-expanded version of s(t) with an instantaneous expansion given by

$$\dot{\theta}(t) = \sqrt{1 + \dot{s}^2(t)}. \tag{3}$$

After being lowpass filtered, the signal is compressed by the system A_o^{-1} in a manner that corresponds exactly in time to the instantaneous expansion performed by the system A_o. This is easily shown by recognizing that the Moore and Parker implementation of the inverse transformation A_o^{-1} requires that

$$t = \theta^{-1}[\theta(t)] \tag{4}$$

and thus

$$\dot{\theta}^{-1}(e) = \frac{1}{\dot{\theta}(t)}, \tag{5}$$

where the reciprocal of $\dot{\theta}^{-1}(e)$ corresponds to the instantaneous compression factor. Consequently, if the impulse response of the lowpass filter has a short duration relative to that of f(e), then the duration of r(t) will be virtually identical to that of s(t).

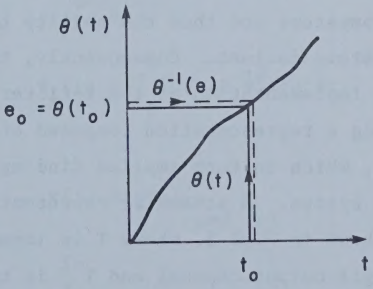


Fig. 3: Mapping reversal used by Moore and Parker in the inverse system A_o^{-1}.

Maintaining an exact temporal correspondence between the signal durations of the input and the output of the E-filter is a fundamental objective of the Moore and Parker system. Since a linear time-invariant filter alone cannot separate signal components whose spectra overlap, a nonlinear time warping is performed to divide the signal spectrally according to its high and low amplitude components. Linear filtering can then be employed to extract the desired component. In returning the resulting signal to the t-domain, processing is performed by A_o^{-1} to invert the time warping applied earlier. The disadvantage of the Moore and Parker approach to the inversion problem is that it requires that the transformation performed by A_o be saved and made available to A_o^{-1}. Hence the system representation of Fig. 1 is misleading in that it implies that successive stages of the system are independent of the operations performed previously. A more explicit representation of their system is shown in Fig. 4 which clearly indicates a channel between A_o and A_o^{-1} containing the time warping transformation.

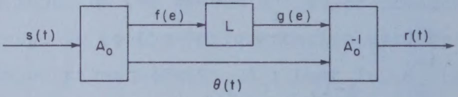


Fig. 4: Explicit representation of the system of Moore and Parker.

AN ALTERNATIVE

An unresolved issue discussed by Moore and Parker is the question of whether or not the E-filter is homomorphic[2]. The difficulty arises essentially because their implementation of the E-filter is a connection of interdependent subsystems and thus the ability to determine the invertibility of the subsystems is lost. Consequently, the problem of finding a homomorphic implementation of the E-filter reduces to the problem of determining a representation composed of independent and invertible subsystems, which in turn implies finding a single-input single-output inverse system. A schematic representation of this system would have the form shown in Fig. 5, where T is identical to the system A_o but has only a single output channel and T^{-1} is the single-channel inverse of T. The single-channel inverse transformation T^{-1} can be derived by considering carefully the nature of the forward transformation. The system T (or A_o) produces f(e) by replotting its input s(t) against its accumulated arc length. The inverting system T^{-1} with input g(e) must therefore have an output r(t) which when plotted against its arc length gives g(e), as shown in Fig. 6.

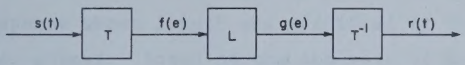


Fig. 5: Single-channel representation of an E-filter.

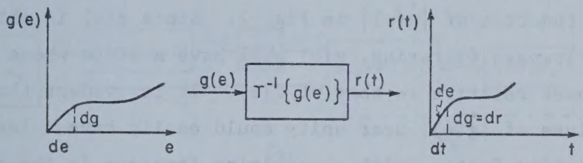


Fig. 6: Single-channel inverse T^{-1}.

Then

$$(de)^2 = (dg)^2 + (dt)^2. \tag{6}$$

Therefore

$$t = \theta_g^{-1}(e) = \int_0^e \sqrt{1 - \dot{g}^2(\alpha)}\, d\alpha. \tag{7}$$

The subscript in $\theta_g^{-1}(e)$ indicates explicitly the dependence on the function g(e). Thus the single-channel system can be described by the relations

$$f(e)\Big|_{e=\theta_s(t)} = s(t) \tag{8a}$$

where

$$\theta_s(t) = \int_0^t \sqrt{1 + \dot{s}^2(\tau)}\, d\tau \tag{8b}$$

and

$$r(t) = g(e)\Big|_{e=\theta_g(t)} \tag{9a}$$

where

$$\theta_g^{-1}(e) = \int_0^e \sqrt{1 - \dot{g}^2(\alpha)}\, d\alpha. \tag{9b}$$

It is important to note that while the single-channel system is guaranteed to be homomorphic, it does not have the property of preserving the correspondence between the durations of its input and output signals. This is due to the fact that the instantaneous compression factor associated with T^{-1},

$$\frac{1}{\dot{\theta}_g^{-1}(e)} = \frac{1}{\sqrt{1 - \dot{g}^2(e)}} , \tag{10}$$

is no longer dependent strictly on f(e). This compression factor is plotted as a function of $|\dot{g}(e)|$ in Fig. 7. Since g(e) is obtained from f(e) by lowpass filtering, g(e) will have a slope whose rate of change is slower relative to that of f(e). It is evident that smoothing affecting values of $|\dot{g}(e)|$ near unity could easily have a large effect on the compression factor, with a resulting increase in the duration of r(t) relative to that of s(t).

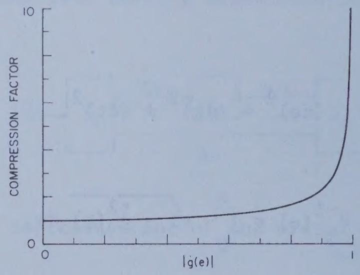


Fig. 7: Compression factor vs $|\dot{g}(e)|$.

EVALUATION

In order to evaluate the performance of the E-filter, a series of simulations of both the Moore and Parker and the single-channel systems was carried out. Whenever possible, the discrete-time algorithms used by Moore and Parker to represent continuous-time operations were implemented. Thus integrals were replaced by summations and derivatives by backward first differences.

Inputs to the systems were samples of various combinations of sinusoids, square waves, and triangular waves. The sampling rate was between 40 and 80 samples per period of the high-amplitude signal. The Moore and Parker condition that the maximum amplitude of the signal be greater than one quarter of the period was also met. Since arc length was approximated by the cumulative distance between sample points, and since these values were generally nonintegers, the function $f(e_n)$ at the output of A_o and T required some form of interpolation before it could be filtered. The method chosen was simple linear interpolation, which Moore and Parker also used. The linear filter was causal with a rectangular impulse response whose length was specified by the amplitude of the high-amplitude signal. Based on the analysis of Moore and Parker this length was chosen to be one-half or less of the maximum amplitude of the input signal. The sequence of operations for both systems is illustrated in Fig. 8.

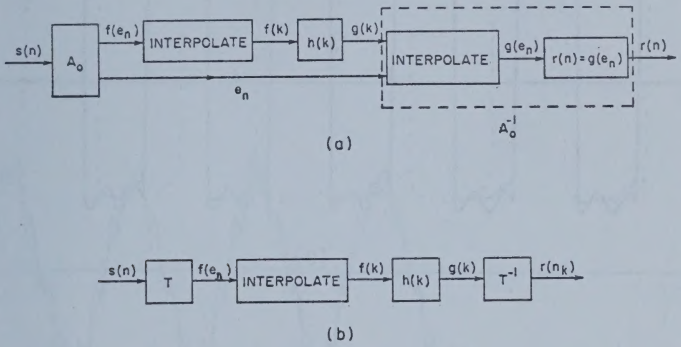


Fig. 8: Discrete-time implementation of E-filters.
(a) System of Moore and Parker.
(b) Single-channel system.

The performances of the Moore and Parker and the single-channel systems were compared for a series of high-amplitude signals with superimposed low-amplitude, high-frequency signals. The results of the computations are shown in Fig. 9, with the time distortion effect of the single-channel system clearly in evidence. It should be pointed out that in the digital implementation of the system signal smoothing is the result of interpolation, as well as of lowpass filtering, and therefore the interpolation performed in the single-channel system must be accomplished at a very high rate to minimize time distortion.

One of the conclusions that may be drawn from the results of the simulations is that the single-channel system has a severe disadvantage when compared with the Moore and Parker system. Any smoothing that takes place within their system will be reflected in smoothing at the output, whereas smoothing within the single-channel system is reflected in smoothing and severe time distortion at its final output. This seems to limit the usefulness of the single-channel system. As far as the

Moore and Parker system is concerned, however, the results of this project show that it appears to have some value as an amplitude filter. Moore and Parker indicate that the filter has been very useful when applied to their work on pattern recognition which requires peak extraction.

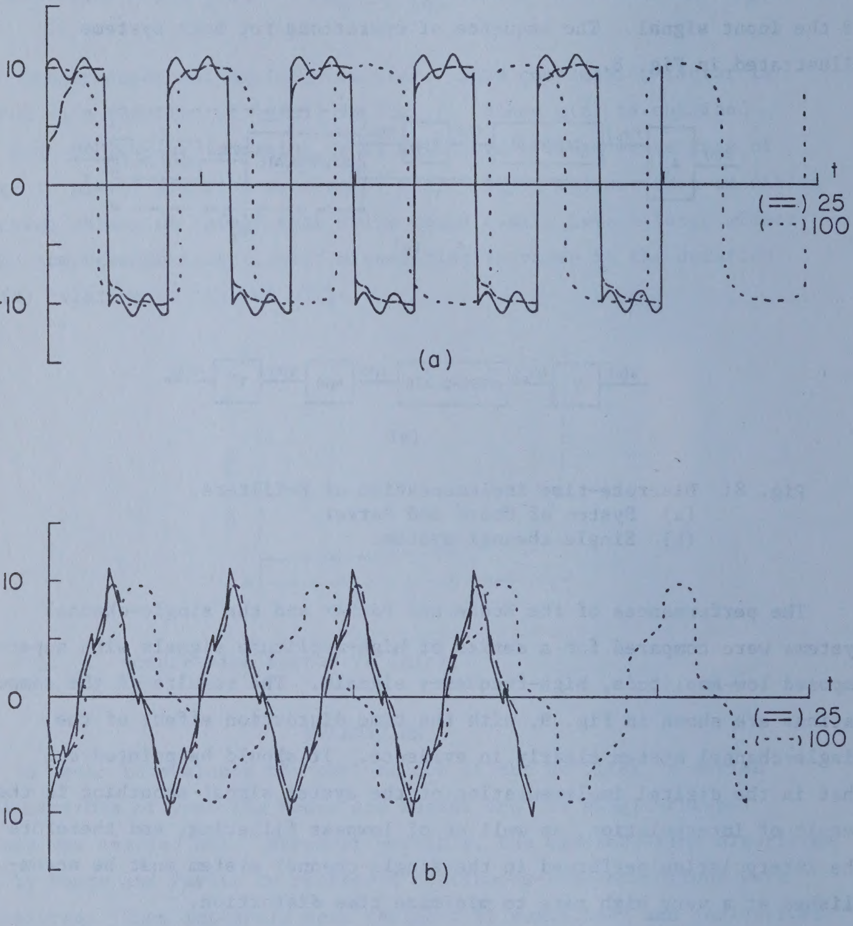


Fig. 9: Comparison of output performance of Moore and Parker system (---) and single-channel system (...) for a sum of signals. Note the change in scale for the single-channel system. Input: (a) square wave + sinusoid, (b) triangular wave + square wave.

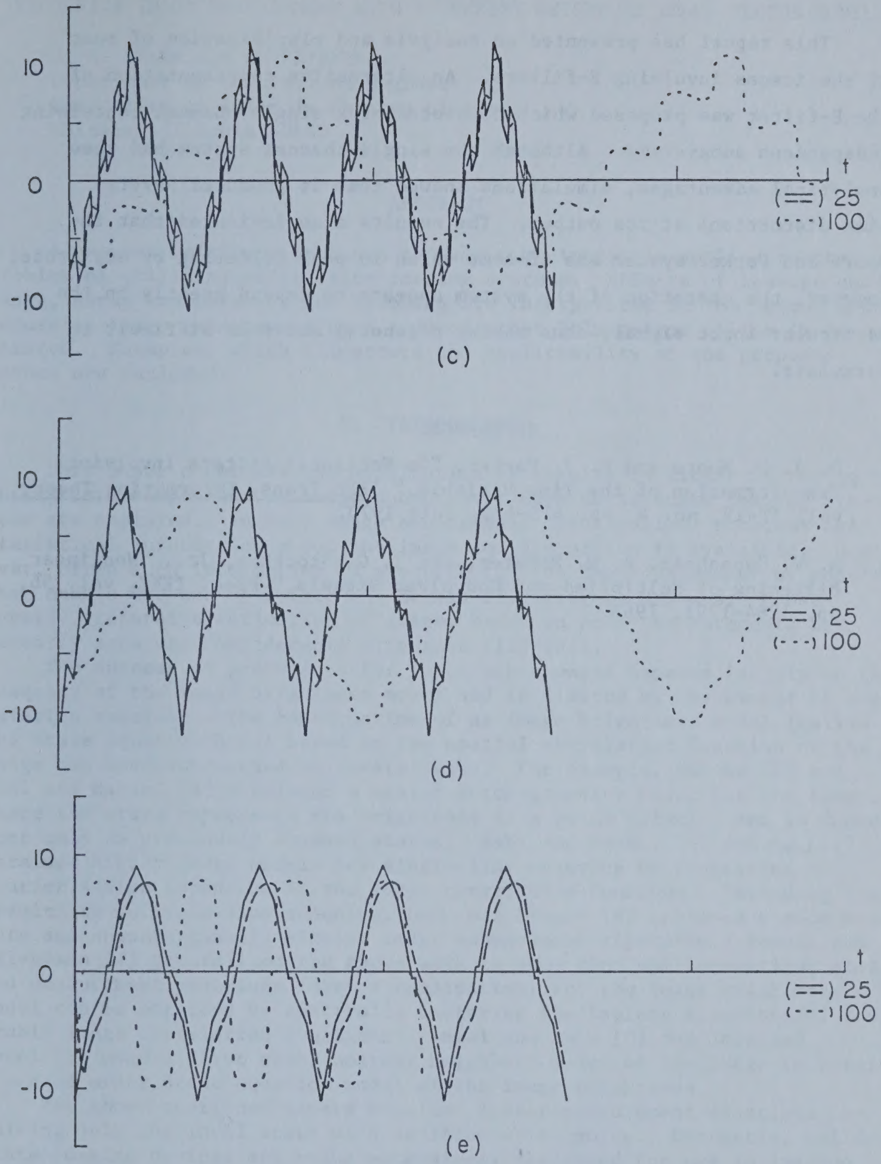


Fig. 9: Comparison of output performance of Moore and Parker system (---) and single-channel system (...) for a sum of signals. Note the change in scale for the single-channel system. Input: (c) triangular wave + square wave, (d) and (e) triangular wave + triangular wave.

SUMMARY

This report has presented an analysis and clarification of some of the issues involving E-filters. An alternative representation of the E-filter was proposed which consisted of a single channel containing independent subsystems. Although the single-channel system had some analytical advantages, simulations showed that it produced severe time distortions at its output. The results also indicated that the Moore and Parker system was of some value in peak filtering by amplitude; however, the operation of the system appears to depend greatly on the particular input signal, thus making a general analysis difficult to formulate.

REFERENCES

1. D. J. H. Moore and D. J. Parker, "On Nonlinear Filters Involving Transformation of the Time Variable," IEEE Trans. Information Theory, vol. IT-19, no. 4, pp. 415-422, July 1973.

2. A. V. Oppenheim, R. W. Schafer, and T. G. Stockham, Jr., "Nonlinear Filtering of Multiplied and Convolved Signals," Proc. IEEE, vol. 56, pp. 1264-1291, 1968.

RECURSIVE IMAGE ENHANCEMENT WITH A NEAREST NEIGHBORS IMAGE SENSOR MODEL

J. C. STAMM and R. PRIEMER
Department of Information Engineering
University of Illinois Chicago Circle
Chicago, Illinois 60680

ABSTRACT

A nearest neighbors approach to modeling images is applied to the problem of utilizing solid state imaging systems. Effects of leakage currents, stray capacitance, and blooming are incorporated in the sensor model. Recursive image enhancement by a reduced order Kalman-type filter is demonstrated. Examples, which illustrate the applicability of the proposed method are included.

I. INTRODUCTION

This paper considers the application of results in linear filtering theory for the problem of image enhancement when solid-state imaging systems are employed. In many image enhancement (restoration) problems only statistical information about the image and distortion is avaialble. However, often such information is adequate [1] for the development of models that enable the use of Weiner and Kalman filtering techniques. Consequently, recursive estimation of images based on noisy measurements has recently received considerable attention ([2]-[9]).

The success of processors for image enhancement depends largely on the adequacy of the image brightness model and is limited by the amount of computation required. The construction of an image brightness model (called the state equation here) based on the spatial correlation function of the image has been approached in several ways. For example, Habibi [2] and Nahi and Habibi [3] developed a scalar autoregressive model for the image, where the state represents the brightness at a point (pixel), and is dependent only on previously scanned states. Nahi and Assefi [4] and Nahi [5] obtained higher order models for single-line scanning by truncating a Fourier series expansion of the image correlation function. Extending these results to multiple-line scanning, Nahi and Franco [6] achieved a more accurate and computationally simpler image enhancement algorithm. Powell and Silverman [7] generalized the above work to show that shift-invariant state and measurement equations (Markov realization) for the image brightness model can be obtained by spectrally factoring the Laplace transform of separable image correlation functions. Angel and Jain [8] and Jain and Angel [9] used a first order nearest neighbors model of the image to obtain a second order state equation model of the image brightness.

The above mentioned papers consider linear measurement equations involving only the local state with additive white noise. Currently, solid-state imaging devices are being extensively developed for use in imaging systems. For such systems the effects of coupling distortion (leakage currents, stray capacitance between neighboring elements, crosstalk along common access lines, and blooming, [10]-[16]) corrupt the measurement along with random minority carrier generation and recombination which can be represented as additive white noise. The present paper develops a flexible model to account for these phenomena and implements this model in a reduced order Kalman filter to recursively estimate an image based on the output of a solid-state imaging system.

II. MODEL CONSTRUCTION

Consider a monochromatic image of dimension (N x N) pixels. Let $z_{n,m}$ represent the brightness of the image at the spatial coordinate (n,m) with n,m = 1, 2, ..., N. A spatially stationary nearest neighbors model (Woods [21]) for the random variable $z_{i,j}$ is

$$z_{i,j} = \sum_I \alpha_{k,\ell} z_{i-k,j-\ell} + \beta u_{i,j}, \quad (k,\ell \neq 0,0) \tag{1}$$

where $u_{i,j}$ is a zero mean uncorrelated random variable and I represents the extent of the image. With this two-dimensional model for the image brightness, the state at any point is correlated to the state of every other point. The coefficients $\alpha_{k,\ell}$ are determined by the simultaneous solution of the set of linear regression equations

$$E[z_{i,j} z_{n,m}] = E[\sum_I \alpha_{k,\ell} z_{i-k,j-\ell} z_{n,m}], \quad (k,\ell \neq 0,0) \tag{2}$$

where n,m = 1, 2, ..., N and $E[\beta u_{i,j} z_{n,m}] = 0$. The coefficient β is found by squaring (1) and taking expectations

$$\beta = \left| \frac{1}{E[u_{i,j}^2]} E[(z_{i,j} - \sum_I \alpha_{k,\ell} z_{i-k,j-\ell})^2] \right|^{\frac{1}{2}}, \quad (k,\ell \neq 0,0) \tag{3}$$

where it is arbitrarily imposed that $E[u_{i,j}^2] = E[z_{i,j}^2]$. Noting that $E[z_{n,m} z_{n+i,m+j}]$ is the autocorrelation function for the image, it will be shown that eqs. (1), (2), and (3) are a basis for a procedure for constructing a Gauss-Markov image brightness model for approximating the statistical nature of a given image. This approach does not require separability or special forms for the image autocorrelation function as in other techniques ([2]-[7]).

Now consider "taking a picture" of an image with a solid-state imaging system. For solid-state image sensors, measurements are possible only at discrete spatial points and represent the intensity of light incident on the local area of photo-sensitive material. The effects of coupling distortion on measurements can be illustrated by a lattice structure joining these points (see Figure 1). For the image sensor, a nearest neighbors model is proposed. The output current $\ell_{i,j}$ at the (i,j) spatial coordinate is

$$\ell_{i,j} = \sum_S d_{k,\ell} z_{i-k,j-\ell} + w_{i,j} \tag{4}$$

where $w_{i,j}$ is zero mean white noise, S represents the spatial sensitivity extent of the image sensor, the $d_{k,\ell}$ $(k,\ell \neq 0,0)$ represent coupling distortion of neighboring elements on the (i,j) element, and $d_{0,0}$ accounts for the contribution of the local incident light intensity to the output.

The magnitude of coupling distortion effects depends on the type and grid size of the sensor, the readout scheme, and the relative intensity of light incident on the sensor. List [17] and Amelio et.al [18] have pointed out the need for eliminating coupling distortion in all types of solid-state imaging systems. Other work, [11], [14], [16], indicates that although these effects can be suppressed, they still significantly contribute to distortion of the sensor output.

It will be shown that the present approach permits the simple construction of high order models and models incorporating specific spatial correlations. Section III shows that any nearest neighbor configuration model is easily accounted for in the filtering algorithm. A first neighbor model for the state equation has been successfully employed by Jain and Angel [9]. The results of Arnold et.al [13] indicate that second neighbor coupling effects account for less than 1% per neighbor of capacitor photodiode type image sensor output, while first neighbors account for about 10% per neighbor. Consequently, a first order model for both the state and measurement equations will be used, i.e.,

$$z_{i,j} = \alpha_{0,-1} z_{i,j+1} + \alpha_{-1,0} z_{i+1,j} + \alpha_{0,1} z_{i,j-1} + \alpha_{1,0} z_{i-1,j} + \beta u_{i,j} \tag{5}$$

$$\ell_{i,j} = d_{0,0} z_{i,j} + d_{0,-1} z_{i,j+1} + d_{-1,0} z_{i+1,j} + d_{0,1} z_{i,j-1} + d_{1,0} z_{i-1,j} + w_{i,j} \tag{6}$$

Define vectors of order N as $x_i = (z_{i,j})$ and $y_i = (\ell_{i,j})$, $j = 1, 2, \ldots, N$ for $i = 1, 2, \ldots, N$. Then (5) and (6) can be written as:

$$x_{i+1} = Q\, x_i - \frac{\alpha_{1,0}}{\alpha_{-1,0}} x_{i-1} + b\, u_i \tag{7}$$

$$y_i = d_{-1,0} x_{i+1} + P\, x_i + d_{1,0} x_{i-1} + w_i \tag{8}$$

where $b = -\beta/\alpha_{-1,0}$ and Q and P are the tridiagonal matrices:

$$Q = \frac{-1}{\alpha_{-1,0}} \begin{bmatrix} 1 & \alpha_{0,-1} & & 0 \\ \alpha_{0,1} & \ddots & \ddots & \\ & \ddots & \ddots & \alpha_{0,-1} \\ 0 & & \alpha_{0,1} & 1 \end{bmatrix}$$

$$P = \begin{bmatrix} d_{0,0} & d_{0,-1} & & 0 \\ d_{0,1} & \ddots & \ddots & \\ & \ddots & \ddots & d_{0,-1} \\ 0 & & d_{0,1} & d_{0,0} \end{bmatrix}$$

Substituting (7) into (8) results in

$$y_i = D_1\, x_i + \left(d_{1,0} - d_{-1,0} \frac{\alpha_{1,0}}{\alpha_{-1,0}} \right) x_{i-1} + v_i \tag{9}$$

where v_i is a random vector of zero mean representing $(d_{-1,0} b\, u_i + w_i)$, $D_1 = d_{-1,0} Q + P$ is a tridiagonal matrix, and D_2, the coefficient of x_{i-1}, is a scalar.

The image enhancement problem now involves a one-dimensional model, and the objective is to estimate the x_i given the actual y_i with (9) as the sensor model. In the next Sect., the application of a reduced order Kalman-type filter to the sytem model (7) and (9) is developed. It is further shown that one-step interpolation can readily be achieved.

III. IMAGE ENHANCEMENT ALGORITHM

Direct application of the Kalman filter to the system model (7) and (9) requires transformation to a representation of order 2N to remove the "delay" terms in the state and measurement equations. However, the computational burden associated with a filter of order 2N may be prohibitive. An alternative is to use the results of [22] and [23] to implement a filter of order N directly for (7) and (9).

Equations similar in structure to the system model (7) and (9) but containing additional "delay" terms result when higher order models (incorporating more nearest neighbors) are constructed. For example, a model involving second order effects in both state and measurement has the form

$$x_{i+2} = Q_1 x_{i+1} + Q_2 x_i + Q_3 x_{i-1} + Q_4 x_{i-2} + b\, u_i$$

$$y_i = D_1 x_i + D_2 x_{i-1} + D_3 x_{i-2} + v_i$$

where Q_2 and D_1 are quindiagonal matrices, Q_1, Q_3 and D_2 are tridiangonal matrices, and Q_4, b and D_3 are scalars. With the results of [22], these more inclusive models can also be employed without further increasing the order of the system.

For completeness, the algorithm for obtaining the MMSE estimate $\hat{x}_{i-\ell/i}$ of $x_{i-\ell}$, $\ell = 0$ and 1 given $y_1, \ldots, y_i$ for the system (7) and (9) is

$$\tilde{y}_{i/i-1} = y_i - D_1\hat{x}_{i/i-1} - D_2\hat{x}_{i-1/i-1} \tag{10}$$

$$\begin{aligned} K_i(\ell) = {} & (V(i-\ell,i/i-1)D_1' + V(i-\ell,i-1/i-1)D_2)(D_1V(i,i/i-1)D_1' \\ & + D_1V(i,i-1/i-1)D_2 + D_2V(i-1,i/i-1)D_1' + D_2^2V(i-1,i-1/i-1) + V_v)^{-1} \end{aligned} \tag{11}$$

$$\hat{x}_{i-\ell/i} = \hat{x}_{i-\ell/i-1} + K_i(\ell)\tilde{y}_{i/i-1} \tag{12}$$

$$V(i-\ell,i-k/i) = V(i-\ell,i-k/i-1) - K_i(\ell)(D_1V(i,i-\ell/i-1) + D_2V(i-1,i-\ell/i-1)) \tag{13}$$

$$V(i-\ell,i+1/i) = V(i-\ell,i/i)Q' - \frac{\alpha_{1,0}}{\alpha_{-1,0}}\, V(i-\ell,i-1/i) \tag{14}$$

$$V(i+1,i+1/i) = QV(i,i+1/i) - \frac{\alpha_{1,0}}{\alpha_{-1,0}}\, V(i-1,i+1/i) + b^2V_u \tag{15}$$

$$\hat{x}_{i+1/i} = Q\hat{x}_{i/i} - \frac{\alpha_{1,0}}{\alpha_{-1,0}}\, \hat{x}_{i-1/i} \tag{16}$$

for ℓ and $k = 0$, 1, and $i = 1, \ldots, N$, where covariances such as $V(n,m/i)$ are defined by

$$V(n,m/i) = E[(x_n - \hat{x}_{n/i})(x_m - \hat{x}_{m/i})']$$

Without any sensor output, the initial condition for the estimator is $\hat{x}_{0/0} = \hat{x}_{1/0} = E[z_{n,m}]$ and covariances such as $V(n,m/0)$ are obtained using the image autocorrelation function.

Examining equations (11) and (12) reveals that obtaining $\hat{x}_{i-1/i}$ (interpolation) only slightly increases the computational burden since the matrix inversion in (11) and most covariances must be computed for filtering. Although the present work does not deal with reducing the computational requirements of this algorithm, the authors obtained considerable reductions in computation time by taking advantage of matrix sparsity and symmetry in multiplication and inversion. One-step interpolation, the processing of y_i to obtain an estimate of x_{i-1}, for the system model (7) and (9) requires $(12N^3 + 20.5N^2 + N)$ multiplications.

To compare the proposed coupling distortion image sensor model to additive noise modeling of the same distortion effects, consider

$$\ell_{i,j} = d_{0,0} z_{i,j} + n_{i,j} \tag{17}$$

$$n_{i,j} = d_{0,-1} z_{i,j+1} + d_{-1,0} z_{i+1,j} + d_{0,1} z_{i,j-1} + d_{1,0} z_{i-1,j} + w_{i,j} \tag{18}$$

Constructing the image sensor model as in Sect. II and assuming the coefficients in (18) are all equal to d results in

$$y_i = d_{0,0} x_i + n_i \tag{19}$$

$$E[n_{i,j}] = 4\,d\,E[z_{i,j}] \tag{20}$$

and $V_n \geq V_w$. Also, the effects of coupling distortion are not "white". Equation (19) was also used as the sensor model along with appropriate filter modifications to account for (20) and V_n.

IV. EXAMPLES

To illustrate the ability of the model (7) and (9) along with the filter (10)-(16) to yield enhancement superior to the use of (19) some examples are now described.

Suppose the image to be considered belongs to the class of images the autocorrelation function of which is

$$E[z_{n,m} z_{n+i,m+j}] = R_{i,j} = \sigma^2 \gamma_1^{|i|} \gamma_2^{|j|} \tag{21}$$

With this symmetric form for $R_{i,j}$, which is stationary and separable, $\alpha_{0,-1} = \alpha_{0,1}$ and $\alpha_{-1,0} = \alpha_{1,0}$ in (5), and the matrix to be inverted in (2) is symmetric. The parameter σ^2 is the average signal power.

The image used is a checkerboard shown in Fig. 2a. To facilitate image display, a line printer was used, and consequently, the number of grey levels (4 here) must be small to enable seeing intensity variations. For this two brightness level image, the light (not white) level was arbitrarily assigned the value 5, and the dark (not black) level the value 1. Therefore, the average image brightness level value is 3, and for $N = 32$ the use of (21) yielded $\sigma^2 = 13.0$, $\gamma_1 = \gamma_2$ (due to image symmetry) $= 0.9404$. In (5), $\alpha_{1,0} = \alpha_{0,1} = 0.2574$ and $\beta = 0.1778$. These parameters specify (7).

For the image sensor and model (6), suppose the $d_{k,\ell}$, $k, \ell \neq 0,0$ are equal to d. To simulate the sensor, a standard uniform distribution white noise generator was used with $V_w = 3.0$ (large enough to make the presence of sensor noise visible with the display mode used).

In example 1, with $d_{0,0} = 1.0$ and $d = 0.125$, (6) was simulated to generate the sensor output in Fig. 2b using the same grey levels as in Fig. 2a. While apparently the contrast increased, actually bright texture is washed out and low brightness sections are dominated by the sensor noise. To more readily see the image error incurred, Fig. 2c is a four grey level display of

$$e_{i,j} = |z_{i,j} - \ell_{i,j}|$$

where $e_{max} = 5.5$. The average power of the image in Fig. 2c is 6.07; therefore the sensor signal to noise ratio is 2.2.

Fig. 3a is the result of obtaining $\hat{x}_{i/i}$ using (10)-(16) with Fig. 2b as the input and (7) and (9) as the system model. Fig. 3b is the zero mean filtering image error using the same grey levels as in Fig. 2c, and the resulting S/N is 10.3. The interpolation error for $\hat{x}_{i-1/i}$ is shown in Fig. 3c with S/N = 19.8.

The parameter values for example 2 are the same as those for example 1, except that (19) was used for the sensor model instead of (9) with

$E[n_{i,j}] = \bar{n}_{i,j} = 1.5$ and

$$V_n = V_w + d^2(4R_{0,0}+8R_{1,1}+2R_{2,0}+2R_{0,2}) - \bar{n}_{i,j}^2 = 3.72$$

Fig. 4 shows the interpolation image error, and the resulting signal to noise ratio is 9.4. The purpose of these two examples is to show that accounting for image sensor coupling effects in the model (9) will yield superior image enhancement results to attributing the coupling distortion to additive sensor noise.

Example 1 was repeated with d = 0.0 to yield the interpolation image error shown in Fig. 5 with S/N = 14.3. This illustrates that coupling effects can represent an advantage over no coupling effects if they are properly accounted for, which is possible with algorithms such as (10)-(16). Some further parameter variations for example 1 and results are shown in Table 1.

V. CONCLUSION

A nearest neighbors model has been proposed in an effort to account for distortion effects resulting with the use of a solid-state image sensor. The model was modified to allow for the recursive processing of N measurements simultaneously. With the coupling distortion model, significant improvements in estimation error variance were achieved as opposed to additive noise modeling of the distortion effects. It was shown that one-step interpolation does not increase the computational burden significantly. The estimation algorithm employed can accomodate most any nearest neighbor model configuration without increasing the dimensionality of the problem.

For high resolution (large dimension) imaging systems, the computational requirements of image enhancement algorithms often are excessive. This aspect was not dealt with here. However, the diagonally oriented nature of the one-dimensional model presented here, along with the diagonal measurement error variance, allows for the decoupling of the model into many lower order parallel systems. Preliminary investigation of a parallel processing image enhancement scheme shows results comparable to the present paper, while considerably reducing the computational requirements of the resulting algorithm. Results are currently under preparation.

Additionally, the sensor model used here may be suitable for representing effects such as sensor jitter and motion degradation. Currently, filters for image enhancement with these models are being investigated, and results will be reported elsewhere.

REFERENCES

[1] Franks, L.E., "A Model for the Random Video Process", Bell System Tech. J., pp. 609-630, April 1966.

[2] Habibi, A., "Two-Dimensional Bayesian Estimate of Images", Proc. IEEE, vol. 60, pp. 878-883, July 1972.

[3] Nahi, N.E. and Habibi, A., "Decision-Directed Recursive Image Enhancement", IEEE Trans. Circuits and Systems, vol. CAS-22, pp. 286-293, March 1975.

[4] Nahi, N.E. and Assefi, T., "Bayesian Recursive Image Estimation", IEEE Trans. Computers, pp. 734-738, July 1972.

[5] Nahi, N.E., "Role of Recursive Estimation in Statistical Image Enhancement", Proc. IEEE, vol. 60, pp. 872-877, July 1972.

[6] Nahi, N.E. and Franco, C.A., "Recursive Image Enhancement-Vector Processing", IEEE Trans. Communications, pp. 305-311, April 1973.

[7] Powell, S.R. and Silverman, L.M., "Modeling of Two-Dimensional Covariance Functions with Application to Image Restoration", IEEE Trans. Autom. Contr., vol. AC-19, pp. 8-12, February 1974.

[8] Angel, E. and Jain, A.K., "A Dimensionality Reducing Model for Distributed Filtering", IEEE Trans Auto. Contr., pp. 59-62, February 1973.

[9] Jain, A.K. and Angel, E., "Image Restoration, Modelling, and Reduction of Dimensionality", IEEE Trans. Computers, pp. 470-476, May 1974.

[10] Osborn, D.C., "Sneak Paths in X-Y Matrix Arrays", IEEE J. Solid-State Circuits, vol. SC-4, pp. 312-317, December 1969.

[11] Anders, R.A., Callahan, D.E., and List, W.F., "Crosstalk in Phototransistor Imaging Mosaics at Very High Light Levels", IEEE J. Solid-State Circuits, vol. SC-4, pp. 326-333, December 1969.

[12] Barbe, D.F., "Noise and Distortion Considerations in Charge-Coupled Devices", Electronics Letters, vol. 8, pp. 207-208, April 1972.

[13] Arnold, E., Crowell, M.H., Geyer, R.D., and Dhirendra, P.M., "Video Signals and Switching Transients in Capacitor-Phototransistor Image Sensors", IEEE Trans. Electron Devices, vol. ED-18, pp. 1003-1010, November 1971.

[14] Sequin, C.H., Shankoff, T.A., and Sealer, D.A., "Measurements on a Charge-Coupled Area Image Sensor with Blooming Suppression", IEEE Trans. Electron Devices, vol. ED-21, pp. 331-341, June 1974.

[15] Mend, W.G., McCoy, E.E., and Anders, R.A., "50 x 50 Solid-State Imaging System Utilizing Hybrid LSI Techniques", Conf. Record, 1970 Solid-State Sensors Symposium, pp. 5-9, Minneapolis, Minn., 1970.

[16] Melen, R.D. and Meindl, J.D., "A Transparent Electrode CCD Image Sensor for a Reading Aid for the Blind", IEEE J. Solid-State Circuits, vol. SC-9, pp. 41-49, April 1974.

[17] List, W.F., "Solid-State Imaging -- Methods of Approach", IEEE Trans. Electron Devices, vol. ED-15, pp. 256-261, April 1968.

[18] Amelio, G.F., Bertram, W.J., and Tompsett, M.F., "Charge-Coupled Imaging Devices - Design Considerations", IEEE Trans. Elec. Devices, vol. ED-18, pp. 986-992, November 1971.

[19] Chien, R.T. and Snyder, W.E., "Hardware for Visual Image Processing," IEEE Trans. Circuits and Systems, vol. CAS-22, pp. 541-551, June 1975.

[20] Biberman, L.M. and Nudelman, S., Photoelectronic Imaging Devices, vol. 2, Plenum Press, New York, 1971. (Chapter 19)

[21] Woods, J.W., "Two-Dimensional Discrete Markovian Fields", IEEE Trans. Info. Thy., vol. IT-18, pp. 232-240, March 1972.

[22] Biswas, K.K. and Mahalanabis, A.K., "Optimal Smoothing for Continuous-Time Systems with Multiple Time Delays", IEEE Trans. Auto. Contr., vol. AC-AC-17, pp. 572-574, August 1972.

[23] Stamm, J.C. and Priemer, R., "Measurement Designs to Improve the Computational Efficiency of Linear System State Estimators", Int. J. Contr., to be published.

[24] Nahi, N.E., Estimation Theory and Applications, John Wiley and Sons, Inc., New York, 1969.

[25] Box, G.E.P. and Jenkins, G.M., Time Series Analysis, Holden-Day, Inc. San Francisco, 1970.

Table I. Example 1 S/N for Sensor Parameter Variations				
$d_{0,0}$	d	Sensor S/N	Filtering S/N	Smoothing S/N
1.00	0.125	2.14	10.29	19.80
1.00	0.0625	3.41	7.45	17.33
1.00	0.00	4.39	4.91	14.22
0.75	0.0625	4.37	4.85	13.11
0.50	0.125	4.23	4.68	12.01

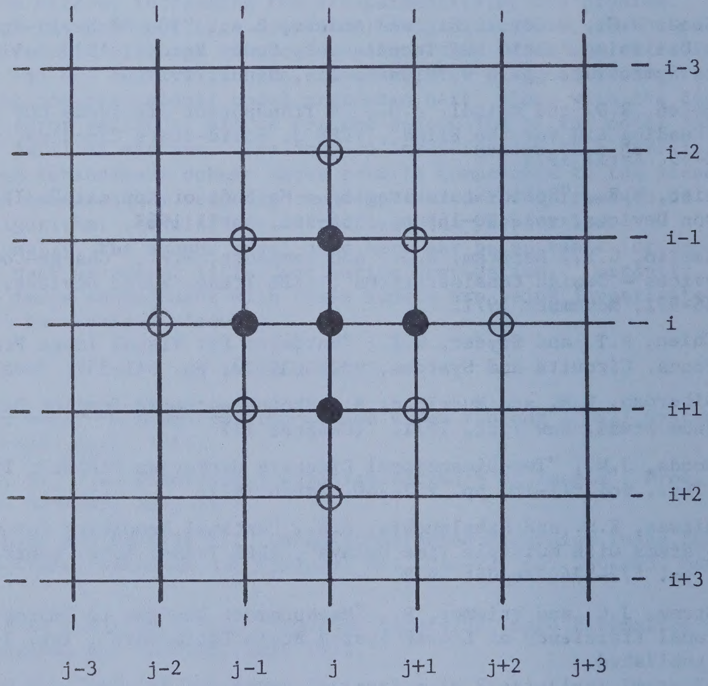


Fig. 1. Lattice structure of image sensor.

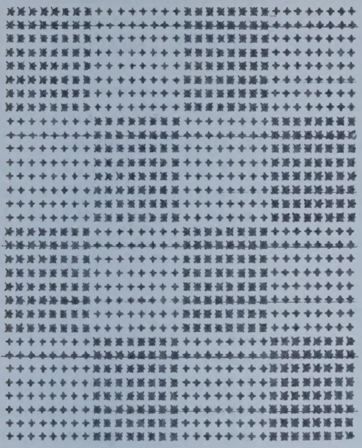


Fig. 2a. Example 1, actual image($z_{i,j}$).

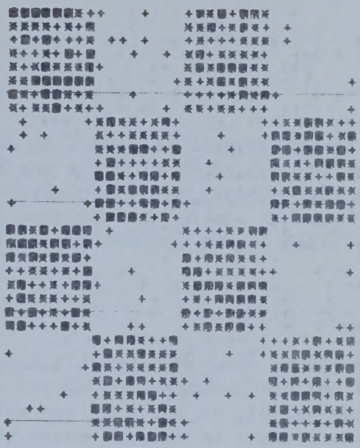


Fig. 2b. Sensor output by eq. (6).

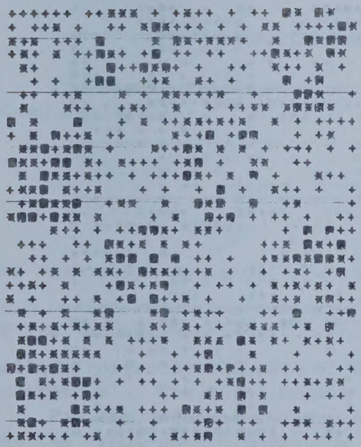


Fig. 2c. Error image (|fig. 2a - fig. 2b|).

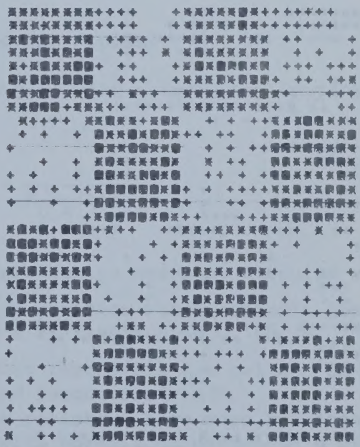


Fig. 3a. Filtering result, $\hat{x}_{i/i}$.

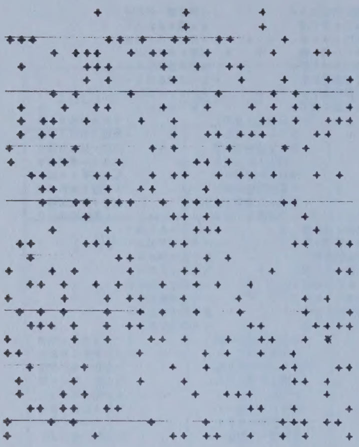


Fig. 3b. Error image (|fig. 2a - fig. 3a|).

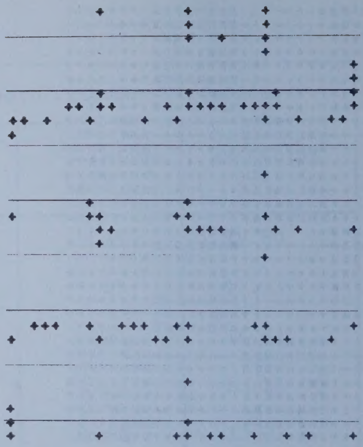


Fig. 3c. Interpolation error image (|fig. 2a - $\hat{x}_{i-1/i}$|).

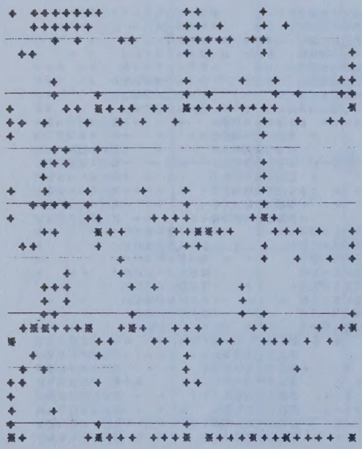


Fig. 4. Example 2 interpolation error image.

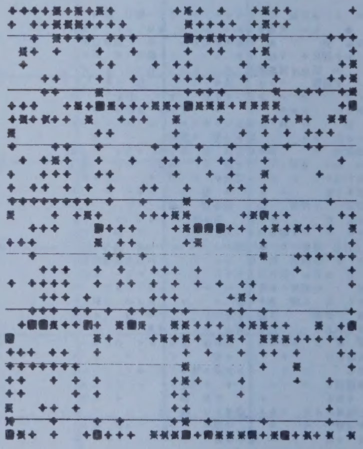


Fig. 5. Example 1 interpolation error image, d=0.0.

A LINEAR PHASE DIGITAL DATA FILTER WITH PREDICTABLE EYE OPENING

H. C. CHAN and C. R. CARTER
Communications Research Laboratory
McMaster University
Hamilton, Ontario, Canada L8S 4L7

ABSTRACT

The performance of the Lerner filter as a digital data filter is investigated using a computer simulation and a simple technique for selecting damping resistors is described. In particular, eye patterns for several different bit rates are presented for both the damped and undamped filters and a formula for predicting eye height for the damped filter is derived.

INTRODUCTION

In the past, Chebyshev and Butterworth filters have been employed as digital data filters due to the low loss inside the passband and the high attenuation outside the passband [1]. Eye patterns can be plotted which graphically illustrate the performance of these filters as transitions produced by typical bit streams are fed to the input. In most cases, however, it is difficult to predict the height of the eye opening whenever the bit rate is near or above the reciprocal of the 3 dB bandwidth due to the 'ringing' transient response which produces intersymbol interference. In this paper, a linear phase bandpass filter, called the Lerner filter [2], is investigated and the results show that, if the filter is damped [3], the height of the eye opening can be predicted over a wide range of bit rates by assuming a simple model for the frequency response. As well, a comparison of the eye patterns for both the undamped and damped Lerner filters shows that the damped filter possesses advantages for obtaining bit timing.

UNDAMPED AND DAMPED LERNER FILTERS

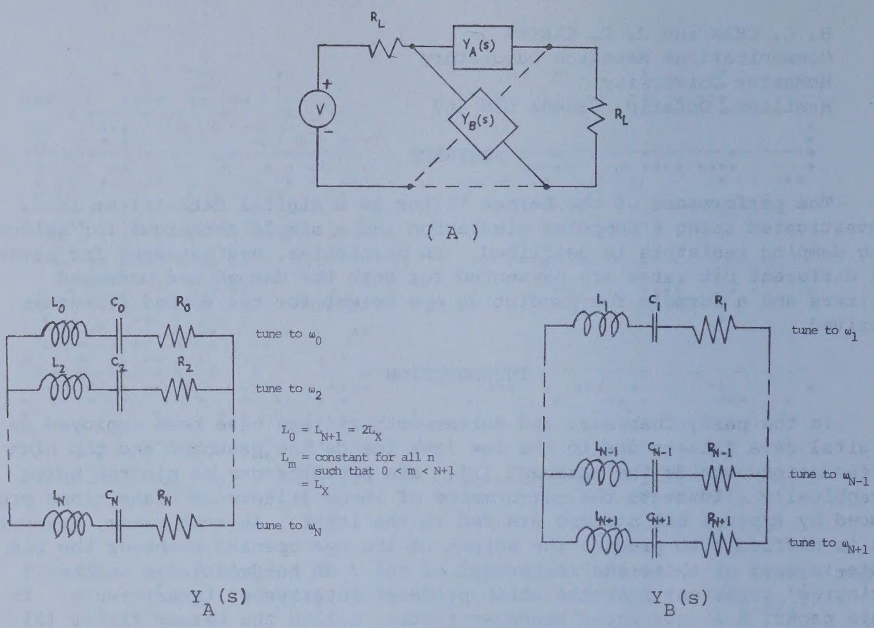
The Lerner filter, illustrated in Fig. 1, has a frequency response given by [2], [3]

$$G(s) = \frac{1}{2}\left[\frac{1}{R_L Y_B(s)+1} - \frac{1}{R_L Y_A(s)+1}\right] \tag{1}$$

The networks comprising $Y_A(s)$ and $Y_B(s)$ are series resonant circuits with each series resonator containing a damping resistor which controls the Q of its respective circuit. The resonators at frequencies ω_0 and ω_{N+1} are called 'corrector resonators' and all other resonators are called 'in-band resonators'. The total number of 'in-band resonators' is N.

For the undamped filter, all damping resistors have zero resistance and the frequency response for a filter with normalized bandwidth B of 3 and normalized center frequency of 15 is shown in Fig. 2, assuming a total of two in-band resonators and two corrector resonators. Eye patterns for this filter for different bit rates are shown in Fig. 3. These patterns were obtained by digital computer simulation using a 127 maximum length pseudo-random sequence.

For the damped filter, a set of resistors is chosen which eliminates



R_L
Y_A(s)
+
V
−
Y_B(s)
R_L
L_0 C_0 R_0
tune to ω_0
L_2 C_2 R_2
tune to ω_2
L_0 = L_N+1 = 2L_X
L_m = constant for all n such that 0 < m < N+1 = L_X
L_N C_N R_N
tune to ω_N
Y_A(s)
L_1 C_1 R_1
tune to ω_1
L_N-1 C_N-1 R_N-1
tune to ω_N-1
L_N+1 C_N+1 R_N+1
tune to ω_N+1
Y_B(s)

(A)

(B)

Fig. 1 (A) Lerner filter, bridge configuration, (B) networks comprising $Y_A(s)$ and $Y_B(s)$ showing the damping resistors.

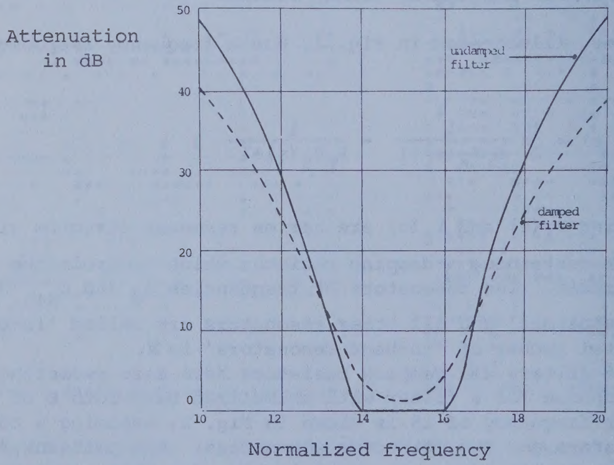


Fig. 2 Frequency response of the undamped and damped Lerner filters.

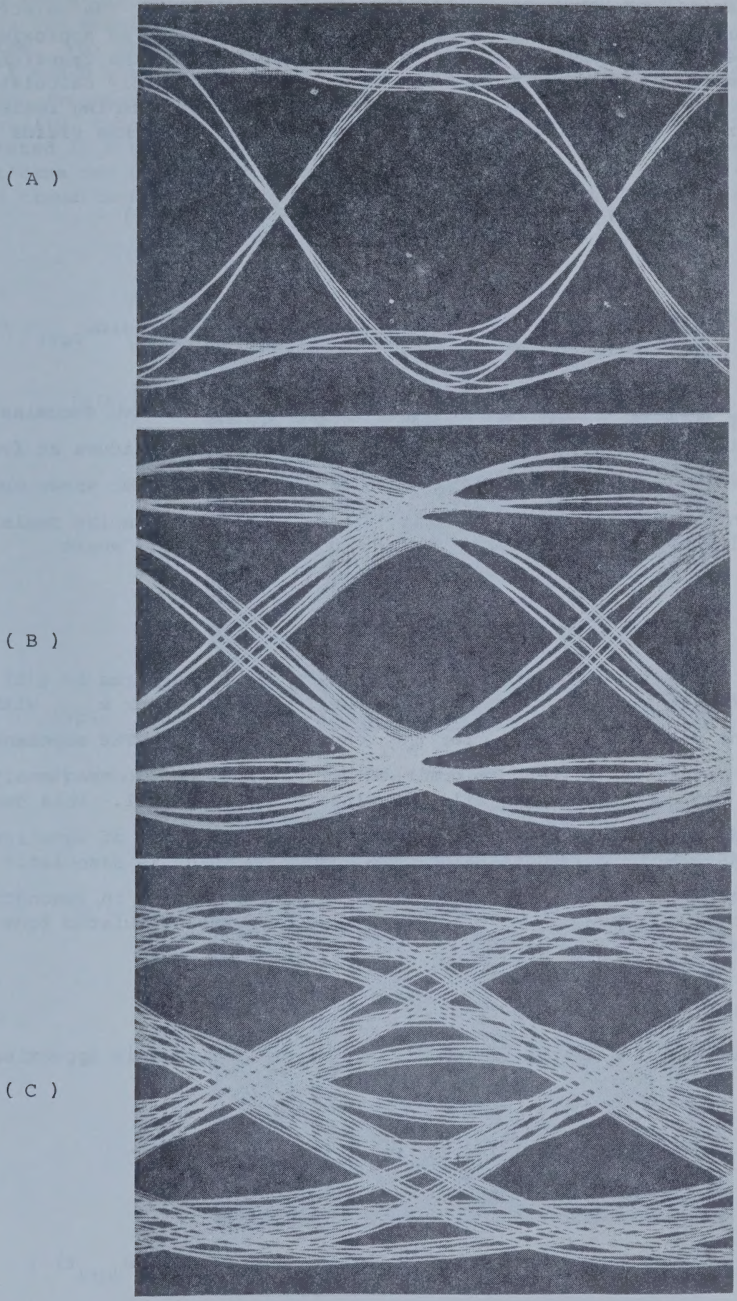


Fig. 3 Eye patterns for the undamped Lerner filter. (A) Bit rate =2, (B) bit rate =3, (C) bit rate =4.

the 'ringing' transient response of the undamped filter. The selection procedure for choosing the damping resistors is based on an approximation developed for calculating the transient response. Now, the transient response of the undamped filter can be determined by simply calculating the inverse Fourier transform of equation (1) with all damping resistors set to zero. For a filter having N in-band resonators, this yields

$$g(t) = \sum_{n=0}^{\frac{1}{2}N} e^{-\alpha_{2n}t}(A_{2n}\cos\omega_{2n}t + B_{2n}\sin\omega_{2n}t) - \sum_{q=0}^{\frac{1}{2}N} e^{-\alpha_{2q+1}t}(C_{2q+1}\cos\omega_{2q+1}t + D_{2q+1}\sin\omega_{2q+1}t) \tag{2}$$

where α_{2n}, ω_{2n} and α_{2q+1} and ω_{2q+1} are the roots of the two denominators in equation (1) and A_{2n}, B_{2n} and C_{2q+1}, D_{2q+1} are the residues at frequencies $\alpha_{2n} + j\omega_{2n}$ and $\alpha_{2q+1} + j\omega_{2q+1}$, respectively. It has been shown that for the i th resonant circuit, the main effect of the i th damping resistor is to simply increase the value of α_i in equation (2) by α_i' where

$$\alpha_i' = \frac{R_i}{2L_i} \tag{3}$$

It is always possible to identify the appropriate terms in g(t) with the different resonators by comparing the frequency ω_{2n} or ω_{2q+1} with the different values of $\sqrt{1/L_iC_i}$ for the resonant circuits. The exponential terms in equation (2) tend to dominate for $t > 2/B$ and, consequently, the fastest damping occurs when all the values of α_i are equal. This can be obtained by calculating the roots of the two denominators of equation (1) and identifying the largest value of α_i, say α_j, which is associated with the j th resonator. Thus, the damping resistor for the j th resonator is taken to be zero and the other damping resistors are calculated from the relation

$$R_i = 2L_i(\alpha_j - \alpha_i) \tag{4}$$

Thus, for the damped filter, the transient response is approximately

$$g'(t) = e^{-\alpha_j t}\left[\sum_{n=0}^{\frac{1}{2}N} (A_{2n}\cos\omega_{2n}t + B_{2n}\sin\omega_{2n}t) - \sum_{q=0}^{\frac{1}{2}N} (C_{2q+1}\cos\omega_{2q+1}t + D_{2q+1}\sin\omega_{2q+1}t)\right] \tag{5}$$

Assuming the filter parameters described above, the values of α_i range from 2.5 to 6.4 and, consequently, $\alpha_j = 6.4$. The effect of using damping resistors is illustrated in Fig. 4. The envelope of each trace represents the magnitude of g(t) or g'(t) and the horizontal axis is normalized time. For the undamped filter, the 'ringing' response continues well beyond t = 2/B whereas the damped filter has negligible response. The resulting frequency response is shown in Fig. 2.

Finally, eye patterns can be plotted for the damped Lerner filter, as illustrated in Fig. 5. Comparing Fig. 3 and Fig. 5, it is seen that the eye patterns for the damped filter are more widely open for high bit rates and the transitions required for bit timing are more precisely defined.

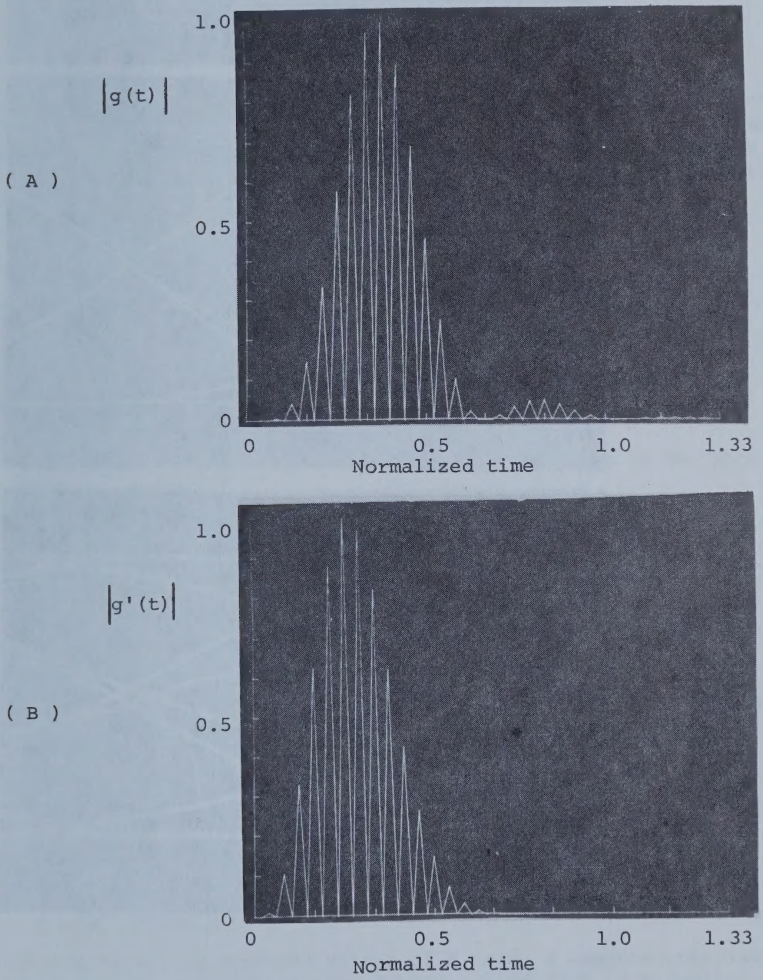


Fig. 4 (A) Transient response of the undamped filter for B = 3.
(B) Transient response of the damped filter.

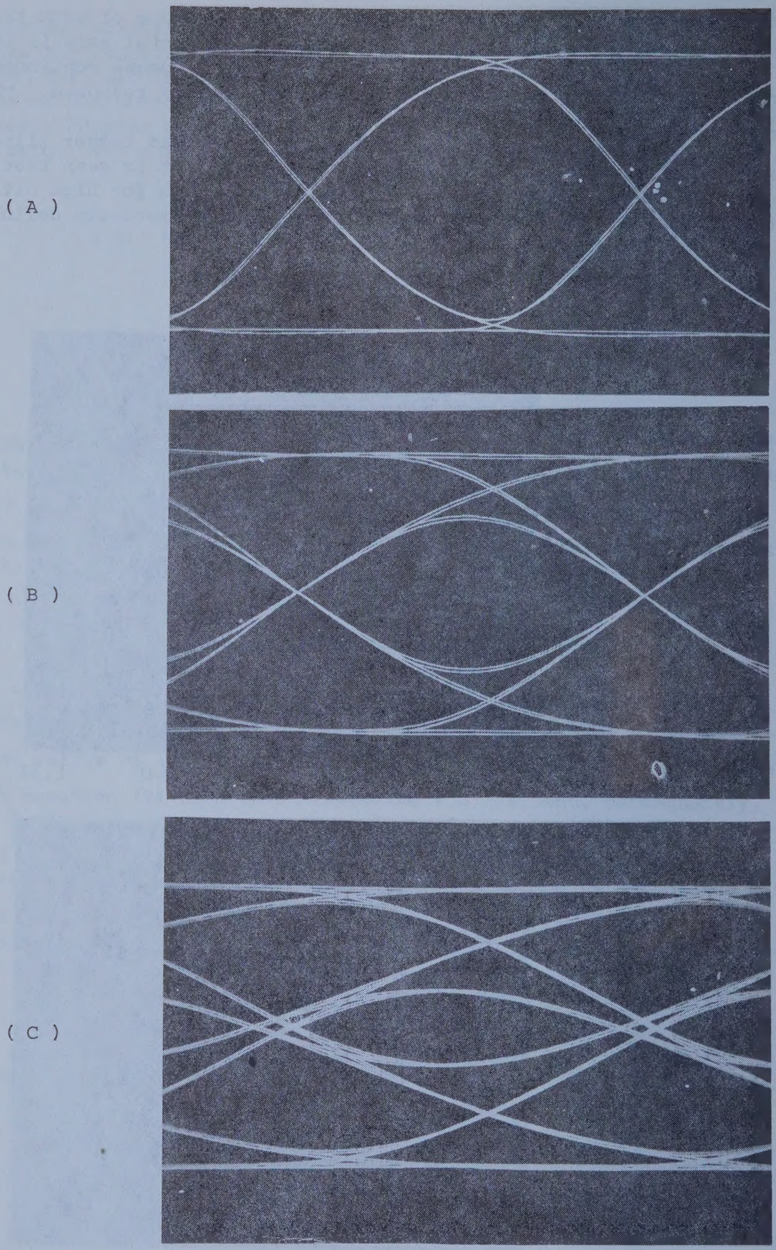


Fig. 5 Eye patterns for the damped Lerner filter. (A) Bit rate =2, (B) bit rate =3, (C) bit rate =4.

EYE HEIGHT FOR THE DAMPED FILTER

Since the problem of intersymbol interference is essentially eliminated by the damped Lerner filter, it is possible to easily calculate the eye height for any data sequence. By inspection, it is seen from Fig. 2 that the baseband frequency response of the damped filter can be approximated by a Gaussian function having the same 3 dB bandwidth. Thus, the voltage transfer function is given by

$$A(f) = 10^{-\frac{3}{5}\left(\frac{f}{B}\right)^2} \qquad (6)$$

The inverse Fourier transform of equation (6) is

$$a(t) = \sqrt{\frac{5\pi}{3\ \ell n\ 10}}\ Be^{-\frac{5}{3\ \ell n\ 10}(\pi Bt)^2} \qquad (7)$$

Now, the smallest eye opening occurs for the data sequences of ---00001000--- and ---11110111---, which can be represented by simply a positive pulse and a negative pulse, respectively. Thus, assume a single pulse, x(t), of unity height and duration T passing through the filter. The output is then

$$y(t,T) = x(t) \otimes a(t)$$

$$= \int_{-\frac{T}{2}}^{\frac{T}{2}} \sqrt{\frac{5\pi}{3\ \ell n\ 10}}\ Be^{-\frac{5}{3\ \ell n\ 10}(\pi B)^2 (t-\tau)^2}\, d\tau \qquad (8)$$

where $\otimes$ indicates convolution.

The maximum of the response occurs when t=0 which is the peak of the eye opening. Hence,

$$y(0,T) = \frac{2}{\pi} \int_0^{\left(\frac{10}{3\ \ell n\ 10}\right)^{\frac{1}{2}} \frac{\pi}{2} BT} e^{-\frac{x^2}{2}}\, dx$$

$$= \operatorname{erf}\left\{\left(\frac{5}{3\ \ell n\ 10}\right)^{\frac{1}{2}} \frac{\pi}{2} BT\right\} \qquad (9)$$

where erf(.) = error function of (.).

From Fig. 6, it is seen that the eye height H is given by

$$H = 2(y(0,T)-0.5)$$

$$= 2\ \operatorname{erf}\left\{\left(\frac{5}{3\ \ell n\ 10}\right)^{\frac{1}{2}} \frac{\pi}{2} BT\right\}-1 \qquad (10)$$

where B = 3 dB bandwidth of the damped filter
T = bit duration

A comparison of measured values of H using a computer simulation with the theoretical prediction provided by equation (10) is illustrated in Fig. 7. The curve is quite general in that it applies for all combinations of B and T.

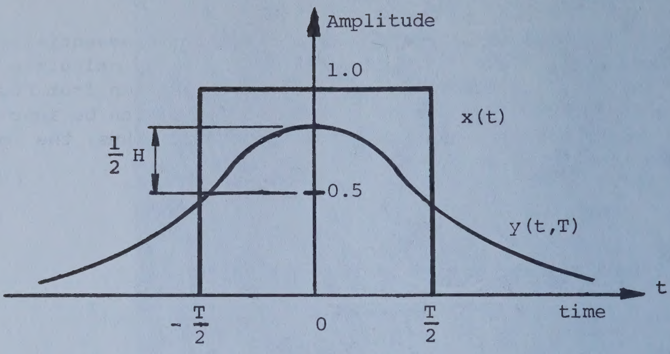


Fig. 6 Eye height for a positive pulse.

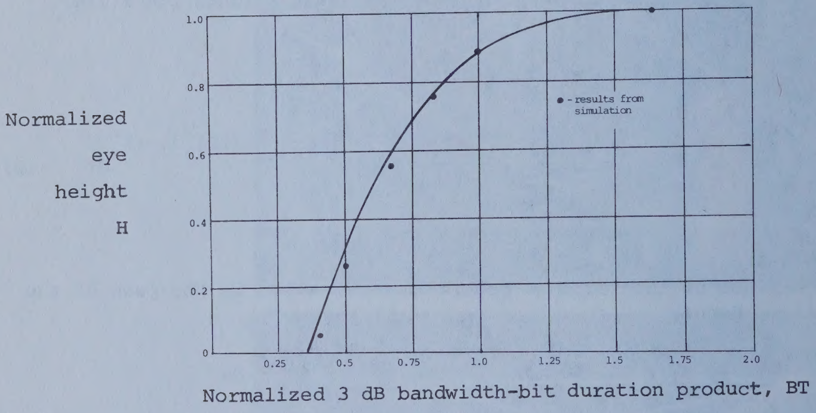


Fig. 7 Normalized eye height versus normalized 3 dB bandwidth-bit duration product.

CONCLUSIONS

A technique for choosing damping resistors for the Lerner filter has been described which greatly reduces intersymbol interference. As a result, it is possible to calculate the eye height using a simple model and simulation results show good agreement with theory.

REFERENCES

[1] Jones, J.J.: 'Filter distortion and ISI effects of PSK signals', IEEE Trans. on Communications, vol. COM-19, pp. 120-132, April 1971.

[2] Lerner, R.M.: 'Band-pass filters with linear phase', Proc. IEEE, March 1964, pp. 249-268.

[3] Carter, C.R.: 'Transient response of the Lerner filter', Proc. 1975 IEEE International Symposium on Circuits and Systems, Newton, Mass., April 21-23, 1975, pp. 36-39.

TWO DESIGN METHODS FOR MINIMUM-PHASE FIR DIGITAL FILTERS

FRANK E. BURRIS
Department of Electrical Engineering & EE Technology
Bradley University
Peoria, Illinois 61625

ABSTRACT

Two design methods for the minimum-phase FIR digital filter, based upon real (or imaginary) part sufficiency and the discrete Hilbert transform {1,2,3,4,7} of the desired log magnitude response of the filter, are presented to complement existing zero-shifting techniques {5,6,7}. In, addition, a brief general discussion of the special character of the minimum-phase response outlines the advantages of such filters.

INTRODUCTION

The current literature mentions several methods for the design of minimum-phase digital filters {5,6,7}. Each of these methods requires the design of a prototype filter having some desired constraint on its magnitude response, the determination of the pole-zero constellation of this prototype, and the alteration of the prototype's pole-zero constellation to achieve a minimum-phase design.

The design methods discussed below take an entirely different approach to the design problem in that the real (or imaginary) part of the desired minimum-phase magnitude response is determined via discrete Hilbert transformation {1,2,3,4,7}. A knowledge of the minimum-phase real (or imaginary) part of the desired frequency response then permits direct determination of the filter coefficients by Fourier analysis or by solution of a set of linear algebraic equations.

REAL (OR IMAGINARY) PART SUFFICIENCY

Consider a general FIR filter H(z),

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} \qquad (1)$$

for which the frequency response function $H(\Theta)$ is given by

$$H(\Theta) = \sum_{n=0}^{N-1} h(n)e^{-jn\Theta} = R(\Theta) + jI(\Theta) \qquad (2)$$

where $\Theta = \omega T$
ω = radian frequency
T = sampling interval (period) of the filter.

The real and imaginary parts ($R(\Theta)$ and $I(\Theta)$) of the frequency response function $H(\Theta)$ are given by the cosine and sine series of (3) and (4), respectively.

$$R(\Theta) = h(0) + \sum_{n=1}^{N-1} h(n) \cos n\Theta \qquad (3)$$

$$I(\Theta) = - \sum_{n=1}^{N-1} h(n) \sin n\Theta \qquad (4)$$

A knowledge of the coefficients h(0), h(1),...., h(N-1) of the cosine series in (3) not only determines the real part R(Θ) of the frequency response but also completely determines the imaginary part I(Θ). It is also apparent that a knowledge of the coefficients -h(1), -h(2),....., -h(N-1) of the sine series in (4) is sufficient to determine the real part R(Θ) to within a constant, h(0).

DESIGN METHOD 1

Development of Method 1

R(Θ), as given by (3), may be interpreted as a truncated Fourier cosine series. This interpretation, in conjunction with the concept of real part sufficiency, leads directly to a design procedure which may be used to design a filter having a prescribed real part of its frequency response. For example, suppose it is desired to design a filter having a desired real part $R_d(\Theta)$. The Fourier series approximation R(Θ) to $R_d(\Theta)$ may be determined and the first N coefficients of R(Θ) may then be used as the coefficients of an N-term FIR filter.

In the context of the minimum-phase filter design problem, the concept of real part sufficiency may be used if the minimum-phase real part is known. Since the filter design problem is usually stated in terms of specifications on the desired magnitude response $|H_d(\Theta)|$, the discrete Hilbert transform may be used to determine the minimum-phase real part $R_d(\Theta)$ from $|H_d(\Theta)|$.

In a similar manner, a Fourier series approximation I(Θ) to a desired imaginary part $I_d(\Theta)$ may be determined and the negatives of the first N-1 Fourier coefficients may be used as the last N-1 coefficients of an N-term filter. In this case, h(0) must be supplied in accordance with some other criterion. Since use of the imaginary part requires an extra step and offers no real advantage over the use of the real part, it will not be discussed further here. It is mentioned only for the sake of completeness and the design procedure outlined below is written in terms of use of the real part only.

Method 1 Procedure Outline

1. Apply a suitable form of the discrete Hilbert transform relation to the desired log magnitude function $A_d(\Theta) = \ln|H_d(\Theta)|$ to obtain the associated minimum-phase function $\phi_d(\Theta)$.

$$\mathcal{H}\{A_d(\Theta)\} = \phi_d(\Theta) \tag{5}$$

2. Use the desired magnitude function $|H_d(\Theta)|$ in conjunction with the derived minimum-phase function $\phi_d(\Theta)$ to obtain the associated real part function $R_d(\Theta)$.

$$R_d(\Theta) = |H_d(\Theta)| \cos \phi_d(\Theta) \tag{6}$$

3. Determine the first N Fourier coefficients (α's) of the cosine series which approximates the even periodic function $R_d(\Theta)$. Call this cosine series R(Θ).

$$R(\Theta) = \alpha_o + \sum_{n=1}^{N-1} \alpha_n \cos n\Theta \tag{7}$$

4. Form the filter $H(z)$ by identifying the N Fourier coefficients of (7) with the N coefficients of the minimum-phase filter $H(z)$.

$$h(n) = \alpha_n \qquad n = 0,1,2,\ldots\ldots\ldots,N-1 \tag{8}$$

$$H(z) = \sum_{n=0}^{N-1} h(n)\, z^{-n} \tag{9}$$

DESIGN METHOD 2

Development of Method 2

The second method to be presented is closely related to the preceding method in that the Hilbert transformation of the desired log magnitude response is utilized to determine the minimum-phase real part of the desired frequency response. However, in this method the Fourier coefficient determination step of Method 1 is replaced by the solution of a set of linear algebraic equations.

Having identified the desired minimum-phase real part $R_d(\Theta)$ of the frequency response, this real part function can then be sampled at a set of discrete frequencies and the sample values can then be used in conjunction with the values of the sampling frequencies and (3) to construct a set of linear algebraic equations which may be solved for the filter coefficients $h(n)$.

The sampling scheme used here is one which, for an N-term filter, samples at least N equally spaced points of the derived $R_d(\Theta)$ over the range of the folding frequency of the filter $(0 \le \Theta \le \pi)$[1]. The discrete frequencies at which $R_d(\Theta)$ is sampled are given by (10).

$$\Theta_i = \omega_i T = \frac{\pi}{N-1}\, i \qquad i = 0,1,2,\ldots\ldots\ldots,N-1 \tag{10}$$

This particular scheme samples the real part function $R_d(\Theta)$ at zero frequency, at the folding frequency, and at N-2 other equally spaced points in between. At each such point Θ_i, the cosine series of (3) becomes a linear equation in N unknowns.

$$R_d(\Theta_i) = h(0) + \sum_{n=1}^{N-1} h(n) \cos n\Theta_i \tag{11}$$

where $R_d(\Theta_i)$, Θ_i, and N are known quantities.

Since there are N sample points Θ_i, a set of N linear equations such as (11) may be written and solved for the N unknown filter coefficients.

---

[1] Certainly this is not the only possible sampling scheme. The number of schemes which might be used to achieve varying objectives is limited only by the user's imagination.

$$\underset{\overline{R}_d}{\begin{bmatrix} R_d(\Theta_0)\\ R_d(\Theta_1)\\ R_d(\Theta_2)\\ \cdot\\ \cdot\\ \cdot\\ R_d(\Theta_{N-1})\end{bmatrix}} = \underset{\overline{E}}{\begin{bmatrix} 1 & \cos\Theta_0 & \cos 2\Theta_0 & \cdots & \cos(N-1)\Theta_0\\ 1 & \cos\Theta_1 & \cos 2\Theta_1 & \cdots & \cos(N-1)\Theta_1\\ 1 & \cos\Theta_2 & \cdots & \cdots & \cos(N-1)\Theta_2\\ \cdot & \cdot & & & \cdot\\ \cdot & \cdot & & & \cdot\\ \cdot & \cdot & & & \cdot\\ 1 & \cos\Theta_{N-1} & \cos 2\Theta_{N-1} & \cdots & \cos(N-1)\Theta_{N-1}\end{bmatrix}} \underset{\overline{h}}{\begin{bmatrix} h(0)\\ h(1)\\ h(2)\\ \cdot\\ \cdot\\ \cdot\\ h(N-1)\end{bmatrix}} \tag{12}$$

(12) may be expressed as a matrix equation $\overline{R}_d = \overline{E}\,\overline{h}$ (13)

where the elements of $\overline{R}_d$ are the known samples, the elements of $\overline{h}$ are the unknowns, and the elements e_{jk} of the NxN coefficient matrix $\overline{E}$ are given by

$$e_{jk} = \cos(k-1)\Theta_{j-1} \qquad \begin{array}{l} j = 1,2,\ldots\ldots,N \\ k = 1,2,\ldots\ldots,N \end{array} \tag{14}$$

The solution of the matrix equation (12) yields the N filter coefficients of the minimum-phase filter.

As in Method 1, an equivalent procedure may be developed using the desired imaginary part $I_d(\Theta)$ and the associated sine series of (4) and a slightly different sampling scheme. As before, the value of h(0) may be chosen in accordance with some other criterion. For the same reasons given in the development of Method 1, the development of Method 2 using the imaginary part of the desired frequency response will not be pursued here.

Method 2 Procedure Outline

1. Apply a suitable form of the Hilbert transform relation to the desired log magnitude function $A_d(\Theta) = \ln|H_d(\Theta)|$ to obtain the associated minimum-phase function at N discrete points Θ_i.

$$\Theta_i = i\,\frac{\pi}{N-1} \qquad i = 0,1,2,\ldots\ldots\ldots,N-1 \tag{15}$$

2. Use the desired magnitude function $|H_d(\Theta)|$ in conjunction with the derived minimum-phase function values $\phi_d(\Theta_i)$ to obtain the associated real part samples $R_d(\Theta_i)$.

$$R_d(\Theta_i) = |H_d(\Theta_i)|\cos\phi_d(\Theta_i) \tag{16}$$

3. Form the set of linear equations of (12) using the Θ_i values of (15) and the $R_d(\Theta_i)$ values of (16).

4. Solve the N linear equations for the N filter coefficients h(0), h(1), h(2),........., h(N-1) and form the minimum-phase filter H(z).

$$H(z) = \sum_{n=0}^{N-1} h(n)\, z^{-n} \tag{17}$$

DESIGN EXAMPLES

For purposes of comparison of the two design methods described above, minimum-phase FIR filters have been designed using a fourth order Butterworth low-pass function as the continuous-time prototype. The desired magnitude response is given by (18).

$$|H_d(\omega)| = \frac{1}{\sqrt{1+\omega^8}} \tag{18}$$

The digital filter to be designed from this prototype has its sampling interval T chosen so that cutoff occurs at 30% of the folding frequency.

$$\Theta_c = \omega_c T = 0.3\pi \tag{19}$$

In each of the design examples, the Hilbert transform operation has been carried out using what the author has called the direct form of the DHT operation {7}, following the approach of Cizek {3}. The discrete Hilbert transform of a sequence is the inverse discrete Fourier transform of the product of the discrete periodic interpretation of the function $j\mathrm{sgn}(\omega)$ and the discrete Fourier transform of the sequence. In the context of the minimum-phase determination problem,

$$\phi(n) = \mathcal{F}_d^{-1}\{E(k)A(k)\} \tag{20}$$

where $\mathcal{F}_d$ = the DFT operation

$A(n)$ = the log magnitude samples of the desired frequency response

$A(k) = \mathcal{F}_d\{A(n)\}$

$$E(k) = \left\{\begin{array}{ll} j & k = 1,2,3,\ldots\ldots,\frac{N}{2}-1 \\ 0 & k = 0,\frac{N}{2} \\ -j & k = \frac{N}{2}+1,\ldots\ldots\ldots,N-1 \end{array}\right\} \quad N \text{ even} \tag{21}$$

$\phi(n)$ = the minimum-phase response sequence.

A 4096-point DHT has been used in each case to design 33-term minimum-phase filters. The 33-term filters designed by Method 1 and Method 2 are denoted $H_1(z)$ and $H_2(z)$ respectively. Each of the resulting designs has been truncated to the minimum number of terms required to meet an arbitrarily chosen maximum error specification of ± 1.0dB over the range of the folding frequency. In each case the minimum number of terms required to meet this error specification on the magnitude response is N = 15. The truncated filters are denoted $H_{1t}(z)$ and $H_{2t}(z)$. Figures 1 and 2 are plots of the magnitude error of both the 33-term and the 15-term filters designed by Methods 1 and 2, respectively. Figures 3 and 4 are plots of the minimum-phase responses of the same filters.

The magnitude error curves for $H_1(z)$ and $H_2(z)$ in Figures 1 and 2 indicate an overall error behavior which is slightly better for $H_1(z)$. This is anticipated since a truncated Fourier series is used in Method 1 to approximate the minimum-phase real part. Thus, the desired real part function is approximated optimally, in the minimum mean square error sense, for a given number of terms.

On the other hand, Method 2 forces the error of the real part approximation to be zero at N equally spaced points over the range of the folding frequency when the number of terms used in the filter is equal to the number of equations solved in the design procedure. This feature of Method 2 provides the designer with some knowledge and control of the magnitude error and insures zero magnitude error at the end points of the frequency range (see the $H_2(z)$ curve of Figure 2). This feature has been used to design a new 15-term filter $H_o(z)$ by solving 15 equations (instead of solving 33 and truncating to 15 terms) using Method 2. The magnitude error curve resulting from this procedure is shown in Figure 5. From this curve it can be seen that the effect of this procedure is to spread the error more evenly over the frequency range and to reduce peak error. Furthermore, error of the real part is forced to be zero at the 15 sample points and any observed error at these points is due entirely to the imaginary part. Thus, Method 2 does provide a certain flexibility in meeting an error specification, a flexibility not provided by Method 1.

Another salient, but important feature of Method 2 is the significant computational savings when compared with Method 1. The discrete Hilbert transform of the desired log magnitude response is required at only N points to supply values of $R_d(\Theta_i)$ to each of the N linear equations. Method 1, on the other hand, requires that a sufficiently large number of points be determined to permit accurate Fourier transformation via the DFT.

The minimum-phase response curves resulting from the two methods (Figures 3 and 4) indicate virtually no difference between the two methods.

SOME ADVANTAGES OF MINIMUM-PHASE DESIGN

In this section, some of the advantages of minimum-phase FIR filter design over linear-phase FIR filters and the continuous-time prototype are briefly highlighted.

Figures 3 and 4 indicate the special nature of the minimum-phase, discrete-time phase response. Over the range of the folding frequency, the net phase shift of the minimum-phase digital filter is zero and at cutoff the phase is only -107.8 degrees. This zero net phase result is not simply a particular result of the examples used here but is a basic characteristic of the minimum-phase, discrete-time response. A linear- phase filter having the same magnitude response and meeting the same magnitude error criterion would exhibit a phase of -648 degrees at the cutoff frequency and a net phase shift of -2160 degrees over the range of the folding frequency. The continuous-time prototype exhibits a phase of -180 degrees at cutoff and a net phase shift in excess of -300 degrees over the range of the folding frequency. Thus, the minimum-phase response represents a very significant savings in system delay over the other two cases mentioned above.

While the minimum-phase response is not linear over the filter's passband, it does exhibit less phase distortion than that of the continuous-time prototype. Phase distortion, as used herein, is defined to be the deviation of the actual phase response from the phase of a straight line drawn from the origin of the phase plot through the point on the actual phase characteristic at the desired cutoff frequency. At $0.3\omega_c$, $0.5\omega_c$, and $0.7\omega_c$ the respective phase distortion values are 8.04, 11.61, and 10.46 degrees for the minimum-phase digital filters and 8.49, 12.04, and 11.11 degrees for the continuous-time prototype.

When compared to the linear-phase filter having the same magnitude response and using a sufficient number of terms to meet the ± 1.0dB error criterion, the minimum-phase filter is found to have a significant economic advantage over the corresponding linear-phase filter. The number of terms required for the minimum-phase filter was found to be 15 whereas the corresponding linear-phase filter requires 25 terms, an increase of 66.7%. This result may be attributed to the optimal (from the standpoint of minimum system delay) clustering of the larger (in magnitude) members of the impulse response sequence of the minimum-phase filter near the time origin.

Further, and much more detailed, arguments for the merits of the minimum-phase digital filter may be found in {7}. It is hoped that the reader has not only discovered several design methods for minimum-phase digital filters, but that an interest in the potential advantages of minimum-phase designs has also been aroused.

REFERENCES

1. A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, Prentice-Hall, 1975, Chapter 7.
2. B. Gold, A.V. Oppenheim, C.M. Rader, "Theory and Implementation of the Discrete Hilbert Transform", Proc. Symp. on Computer Processing in Communications, Vol. 19, Polytechnic Press, 1970, New York.
3. V. Cizek, "Discrete Hilbert Transform", IEEE Trans. on Audio and Electroacoustics, Vol. AU-18, No. 4, 1970, pp. 340-343.
4. F. Bonzanigo, "A Note on 'Discrete Hilbert Transform'", IEEE Trans. on Audio and Electroacoustics, (correspondence), Vol. AU-20, No. 1, 1972, pp. 99-100.
5. O. Herrmann and H.W. Schuessler, "Design of Nonrecursive Digital Filters with Minimum Phase", Electronics Letters, Vol.6, No. 11, 1970, pp. 329-330.
6. F.E. Burris, "Design of Minimum-Phase Digital Filters from Linear-Phase Prototypes", Electronics Letters, Vol.10, No. 6, 1974, pp. 69-70.
7. F.E. Burris, "Phase Control and Phase Minimization in Digital Filters", Ph.D. Dissertation, Univ. of Cincinnati, Cincinnati, OH, 1974.

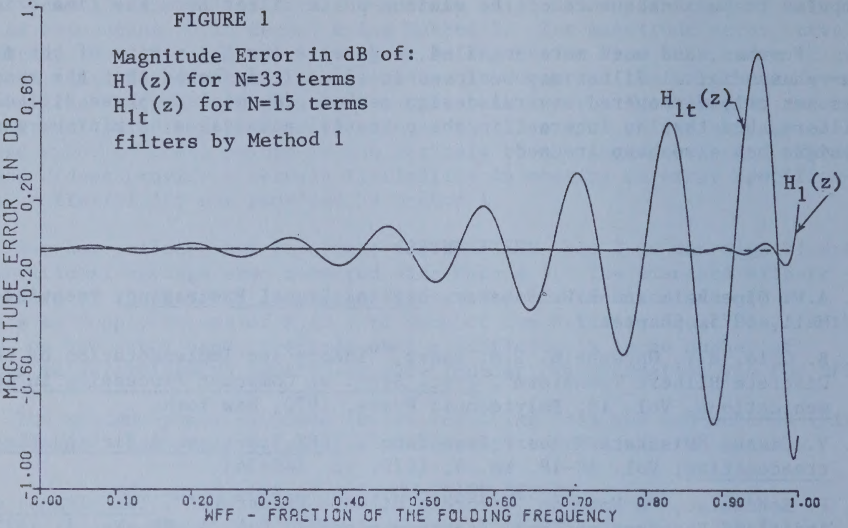


FIGURE 1

Magnitude Error in dB of:
$H_1(z)$ for N=33 terms
$H_{1t}(z)$ for N=15 terms
filters by Method 1

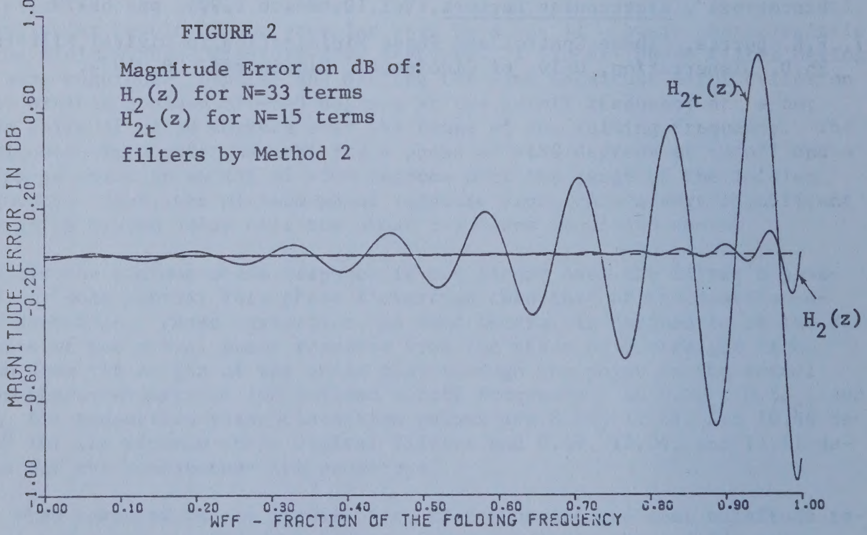


FIGURE 2

Magnitude Error on dB of:
$H_2(z)$ for N=33 terms
$H_{2t}(z)$ for N=15 terms
filters by Method 2

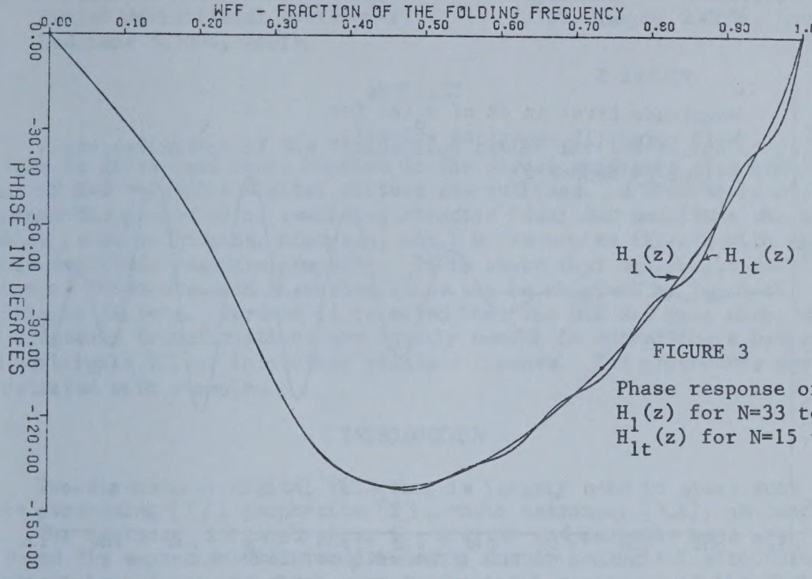


FIGURE 3

Phase response of:
$H_1(z)$ for N=33 terms
$H_{1t}(z)$ for N=15 terms

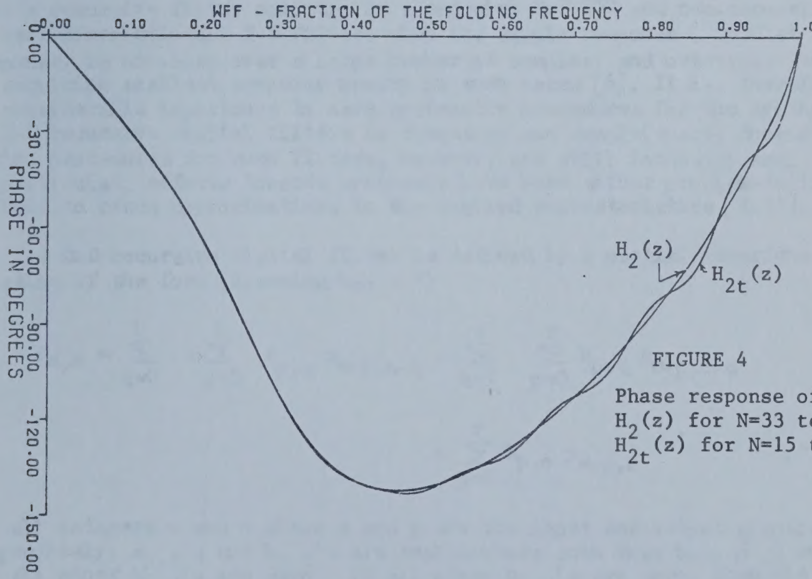


FIGURE 4

Phase response of:
$H_2(z)$ for N=33 terms
$H_{2t}(z)$ for N=15 terms

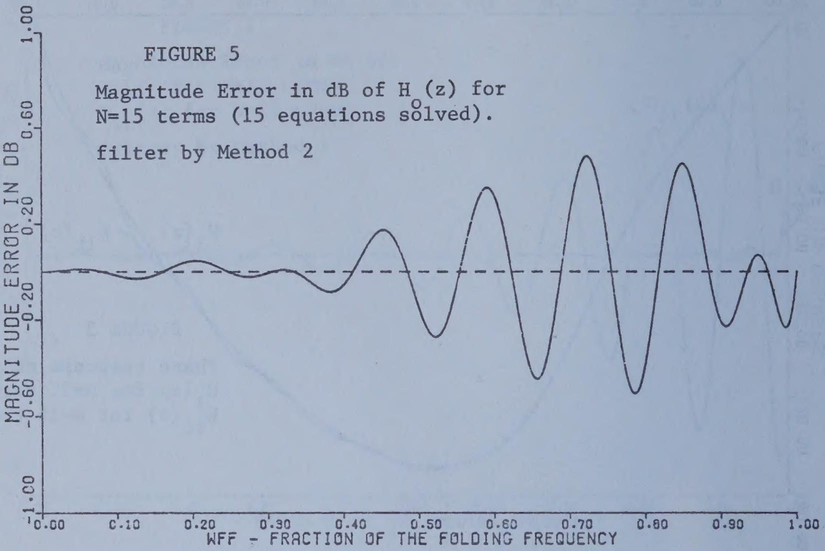


FIGURE 5

Magnitude Error in dB of $H_o(z)$ for N=15 terms (15 equations solved).

filter by Method 2

SYNTHESIS OF EQUIRIPPLE RECURSIVE TWO-DIMENSIONAL DIGITAL FILTER

H. S. SEKHON
Department of Electrical Engg.
Punjab Agricultural University,
Ludhiana 141004, India.

V.P. SINHA
Department of Electrical Engg.
I.I.T. Kanpur 208016, India.

ABSTRACT

A new definition of the "transition ratio" for the design of 2-D filters is given; and basic hurdles in the direct synthesis of a general class of 2-D recursive digital filters are outlined. A theorem is stated to hinge the procedure of realizing standard ideal 2-D amplitude characteristics (such as lowpass, highpass, etc.) by recursive filters with equiripple amplitude response property. It is shown that equiripple realization of these standard characteristics can be obtained by two 1-D equiripple filters. Further it is noted that for the 2-D case also, the 1-D frequency transformations are equally useful in converting a lowpass 2-D equiripple filter into other standard classes. The procedures are illustrated with examples.

INTRODUCTION

Two-dimensional digital filtering is largely used in areas such as image processing [1], geophysics [2], radio astronomy [3,4], photometry[4] etc. For instance, in geophysics, the gravity and magnetic maps are filtered "to emphasize features presumably due to geological structures of practical importance and clear away the regional component of the field"[5].

In the past, such spectrum shaping operations have mainly been carried out with the help of the Fourier transform techniques, the latest of these being the 2-D fast Fourier transform. Nevertheless, it is seen that a recursive filter competes in speed with 2-D FFT and non-recursive (direct-convolutional) 2-D filters when the sample response (impulse response) is non-zero over a large number of samples; and outstrips these in requiring smallest computer memory in such cases [6]. It is, therefore, of considerable importance to have systematic procedures for the synthesis of 2-D recursive digital filters in frequency and sample(space) domains. Design procedures for such filters, however, are still inchoate; and, in particular, efforts towards synthesis have been either problem-tailored or lead to crude approximations to the desired characteristics [7-11].

A 2-D recursive digital filter is defined by a partial recurrence relation of the form (assuming $b_{oo} = 1$)

$$y_{m,n} = \sum_{q=0}^{l} \sum_{p=0}^{k} a_{p,q}\, x_{m-p,n-q} - \sum_{q=1}^{s} \sum_{p=0}^{r} b_{p,q}\, y_{m-p,n-q} - \sum_{p=1}^{r} b_{p,o}\, y_{m-p,n} \qquad (1)$$

for all integers m and n where x and y are the input and output signals respectively; $a_{p,q}$'s and $b_{p,q}$'s are real numbers such that $b_{o,o} \neq 0$ and not all other $b_{p,q}$'s are zero. If all other $b_{p,q}$'s are zero, then the filters are called non-recursive filters. Considering the input and output signals to be zero for negative m or n the transfer function $H(z_1, z_2)$ of the filter in (1) can be obtained by taking 2-D z-transform [13] of both sides of (1) as

$$H(z_1,z_2) = \frac{\sum_{q=0}^{l} \sum_{p=0}^{k} a_{p,q}\, z_1^{-p}\, z_2^{-q}}{\sum_{q=0}^{s} \sum_{p=0}^{r} b_{p,q}\, z_1^{-p}\, z_2^{-q}} \qquad (2)$$

where z_1 and z_2 are z-transform variables. Frequency responses of the filter can be obtained by substituting $z_1 = \exp(jw_1)$ and $z_2 = \exp(jw_2)$, where w_1 and w_2 are relative or normalized frequency variables, in its transfer function $H(z_1, z_2)$. The squared amplitude response, $G(w_1,w_2)$, of such filters can be expressed in the explicit form [14] as

$$G^2(w_1,w_2) = \frac{\sum_{u_2=-l}^{l} \sum_{u_1=-k}^{k} C_{u_1,u_2} \cos(u_1w_1 + u_2w_2)}{\sum_{u_2=-s}^{s} \sum_{u_1=-r}^{r} D_{u_1,u_2} \cos(u_1w_1 + u_2w_2)} \qquad (3)$$

where C_{u_1,u_2} and D_{u_1,u_2} are real numbers. It can be observed from (3) that $G(w_1,w_2)$ is doubly periodic with period 2π. So it is enough to concentrate on a particular band of frequencies like $|w_1| \leq \pi$ and $|w_2| \leq \pi$. The band defined by these frequencies is termed as the baseband.

DEFINITIONS

1. 2-D Ideal Amplitude Characteristics:

There are four broad classes of ideal 2-D filters: (a) lowpass, (b) highpass, (c) bandpass and (d) bandelimination. All these characteristics are exhibited in Fig.1.

2. Transition Band:

Amplitude response of most of the commonly used filters is composed of pass and stop bands. In the passband, the attenuation is less than a prescribed value, while in the stopband the attenuation is greater than a given value. Owing to the realizability constraints which do not permit abrupt transitions, pass and stop bands are conjoined by a buffer band known as the transition band. The two boundaries that mark a transition band, called the cut-off and transition boundaries, can formally be defined as follows:

Cut-off Boundary: The cut-off boundary is defined as the set of frequency points (w_1,w_2), known as cut-off frequencies, whose every neighbourhood contains a point where attenuation is greater than the passband attenuation, and it also contains a point where attenuation is less than or equal to the attenuation in the passband.

Transition Boundary: The transition boundary is defined as the set of frequency points, known as the transition frequencies, whose every neighbourhood contains a point where attenuation is less than the stop-band attenuation and it also contains a point where the attenuation is greater than or equal to the attenuation in the stop-band. These are illustrated in Fig.2 for the idealized case of a 2-D digital filter.

3. Separable Characteristics:

A given 2-D amplitude characteristic, $G(w_1,w_2)$, is said to be separable if it can be written as product of two 1-D characteristics, i.e. $G(w_1,w_2) = g_1(w_1)\, g_2(w_2)$ where g_1 and g_2 are 1-D amplitude characteristics.

4. Transition Ratio:

The transition ratio of a filter is a measure of sharpness of the change of the amplitude response from passband to the stopband and vice-versa. In 1-D case, the transition bands are intervals and, therefore, for the lowpass filter, the transition ratio is simply defined as Cut-off frequency/Transition frequency. But, in the case of 2-D filters, the transition bands being regions, a meaningful definition for a 2-D lowpass filter can be given as

$$\left[\frac{\text{Passband area}}{\text{Passband area + Transition band area}}\right]^{\frac{1}{2}}.$$

To see how much 'meaningful' this definition is, consider a 2-D lowpass filter with cut-off frequencies w_{c1} & w_{c2} along the w_1 and w_2 axis respectively; and let w_{T1} and w_{T2} be the corresponding transition frequencies. Then, assuming the cut-off boundary and transition boundary to be rectangular (which is not too much of an assumption really), the transition ratio becomes

$$\left[\frac{(2w_{c1})(2w_{c2})}{(2w_{T1})(2w_{T2})}\right]^{\frac{1}{2}} = \left[\frac{w_{c1}}{w_{T1}}\right]^{\frac{1}{2}} \cdot \left[\frac{w_{c2}}{w_{T2}}\right]^{\frac{1}{2}} = (\text{Transition ratio of 1-D lowpass } w_1\text{-axis filter})^{\frac{1}{2}} \times (\text{Transition ratio of 1-D lowpass } w_2\text{-axis filter})^{\frac{1}{2}}.$$

This shows that transition ratio requirement for a 2-D lowpass filter can be met by two 1-D lowpass filters with appropriate transition ratios. The same concept can similarly be extended to other classes of 2-D filters and need not be elaborated here.

5. Equiripple Functions:

Let $f(x)$ be a real-valued function of a real variable x defined over the closed interval, $a \leq x \leq b$. It is said to be equiripple if its global max is equal to its local maxima; and global min. is equal to its local minima. Similarly, a real-valued function, $f(x,y)$, of two real variables x and y defined over the closed rectangle $a \leq x \leq b$ and $c \leq y \leq d$, is said to be equiripple over the rectangular region if its global max is equal to its local maxima; and global min. is equal to its local minima.

6. Equiripple Filter:

A filter with amplitude response, f, is said to be approximate a given amplitude response, g, in the equiripple sense(and is briefly referred to as an equiripple filter), if the error $e = g - f$ is an equiripple function. If the given characteristic can be divided into pass and stop bands with constant response, then f itself becomes an equiripple function over these bands. Further, for a meaningful approximation, the passband and stopband responses being assumed to be 1 and 0 respectively, the absolute error in each of these bands should be $\ll 1$.

THE SYNTHESIS PROBLEM

Let us recall that the squared amplitude response of a 2-D recursive digital filter is a generalized rational function of cosine functions in two variables. Unfortunately, the theory of approximation with such generalized rational functions of two variables is in an early stage and no satisfactory algorithms are available so far for arriving at approximations of this form [15]. Further, even if it will be possible to obtain a function of the form (3), the next step of constructing from it a stable 2-D transfer function, since the fundamental theorem of algebra does not extend to two variables [1], would be equally difficult.

In view of the above mentioned hurdles in the direct synthesis of 2-D recursive filters through the form(3), we have to look for alternative strategies. It is in this context that we will explore a special class of 2-D recursive filters - the class of separable filters. We shall presently see that most of the ideal filter requirements can be met with the help of this class of 2-D filters. In particular, the standard characteristics, i.e. the lowpass, highpass, bandpass and bandelimination characteristics, can always be realised by separable recursive filters.

SYNTHESIS OF STANDARD FILTERS

Procedures for designing such filters depend mainly upon a simple but useful property of equiripple functions and their products, which is stated below as a lemma. With its help we will show that the problem of synthesizing 2-D equiripple filters can be resolved into that of designing two 1-D equiripple filters.

Lemma : Let $f_1(x)$ and $f_2(y)$ be two real-valued non-negative functions of real variables x and y defined over the closed intervals $a \leq x \leq b$ and $c \leq y \leq d$ respectively, such that neigher f_1 nor f_2 remains constant over any interval of its domain, howsoever small. Then the function, $f(x,y) = f_1(x) f_2(y)$, is equiripple over the closed rectangle $a \leq x \leq b$, $c \leq y \leq d$ iff $f_1(x)$ and $f_2(y)$ are equiripple functions.

Proof: The proof is given in [14].

We shall now see how the result of Lemma can be used for the synthesis of such filters to give equiripple amplitude response in the passband and stopband. We shall first take up the synthesis of a lowpass filter. We begin by observing that the ideal 2-D lowpass characteristics are separable. Thus, if $G(w_1,w_2)$ denotes the desired ideal 2-D lowpass amplitude response, then it may be split up as $G(w_1,w_2) = g_1(w_1)\ g_2(w_2)$ where

$$g_1(w_1) = \begin{cases} 1 \text{ for } |w_1| \leq w_{c1} \\ 0 \text{ for } w_{c1} < |w_1| \leq \pi \end{cases} \text{ and } \quad g_2(w_2) = \begin{cases} 1 \text{ for } |w_2| \leq w_{c2} \\ 0 \text{ for } w_{c2} < |w_2| \leq \pi . \end{cases}$$

From this we see that g_1 and g_2 may be regarded as ideal 1-D lowpass amplitude characteristics. We shall consider the equiripple approximation of $G(w_1,w_2)$ by a separable recursive filter with amplitude response of the form $F(w_1,w_2) = f_1(w_1)f_2(w_2)$ where $f_1(w_1)$ and $f_2(w_2)$ are the amplitude responses of two 1-D recursive filters. The filter with amplitude response $f_1(w_1)$ $f_2(w_2)$ is referred to as the w_1-axis(w_2-axis) filter. Our requirements for approximation are: (i) $F(w_1,w_2)=f_1(w_1)f_2(w_2)$ should approximate $G(w_1,w_2)$, over the pass and stop bands, in the equiripple sense, which is equivalent to having $F(w_1,w_2)$ itself an equiripple function; and (ii) for the approximation to have any import the passband and stopband ripples should be considerably less than one. Since f_1 and f_2 are amplitude responses of 1-D recursive filters, they are non-negative; and are non-constant over any interval of their domains $|w_1| \leq \pi$ and $|w_2| \leq \pi$ respectively, howsoever small. By Lemma, requirement (i) necessitates that each of f_1 and f_2 be equiripple. As to the requirement (ii) we consider the cases of passband and stopband separately. First concentrating on the passband, we note that if we have

$$\min_{|w_1| \leq w_{c1}} f_1(w_1) = 1 - r_{p1} \text{ and } \min_{|w_2| \leq w_{c2}} f_2(w_2) = 1 - r_{p2}$$

where $r_{p1}, r_{p2} \geq 0$, then the passband ripple, r_p is given by

$$r_p = 1 - \min_{\substack{|w_1| \leq w_{c1} \\ \text{and} |w_2| \leq w_{c2}}} F(w_1,w_2) = 1-(1-r_{p1})(1-r_{p2}) = r_{p1} + r_{p2} - r_{p1}\, r_{p2} \quad (6)$$

From (6) we may observe that r_p can be made sufficiently small by choosing smaller r_{p1} and r_{p2}. Similarly, for the stopband, if

$$\max_{w_{T1} \leq |w_1| \leq \pi} f_1(w_1) = r_{s1} \text{ and } \max_{w_{T2} \leq |w_2| \leq \pi} f_2(w_2) = r_{s2}$$

where $w_{T1} \geq w_{c1} + \epsilon_1$ and $w_{T2} \geq w_{c2} + \epsilon_2$, $\epsilon_1, \epsilon_2 > 0$ (w_{T1} and w_{T2} may be recognized as the transition frequencies of the respective 1-D filters), then the stopband ripple, r_s, is given by $r_s = \max f_1(w_1)f_2(w_2) = r_{s1}$ or r_{s2} whichever is larger over the stopband. (7)

From (7) we see that the stopband ripple, r_s can be made sufficiently small by choosing small r_{s1} and r_{s2}. All these results can be summarised in a theorem stated below.

Theorem 1: The problem of synthesizing an equiripple lowpass 2-D recursive digital filter is equivalent to that of synthesizing two 1-D recursive lowpass filters. The passband and stopband ripples of the 2-D filter are related to those of the 1-D filters as follows:

$$r_p = r_{p1} + r_{p2} - r_{p1}\, r_{p2} \text{ and } r_s = r_{s1} \text{ or } r_{s2} \text{ whichever is larger.}$$

Example 1: Synthesize a 2-D lowpass recursive digital filter with equiripple amplitude response to meet the following requirements: Cut-off frequency along the w_1-axis $= w_{c1} = 0.4$ radians/sample, Cut-off frequency along the w_2-axis $= w_{c2} = 0.4$ radians/sample, Transition ratio ≥ 0.8, Passband ripple $= r_p \leq 0.0975$ and Stopband ripple $= r_s \leq 0.01$.

Solution: By Theorem 1, we require the synthesis of two 1-D recursive filters with equiripple amplitude responses. For convenience, we will have identical 1-D lowpass filters each having the following specifications : Cut-off frequency $= 0.4$ radians/sample, Transition ratio(from Definition) ≥ 0.8, Passband ripple ≤ 0.05, Stopband ripple ≤ 0.01. The 1-D lowpass filter with above specifications was designed using the available digital filter synthesis program [16] and its transfer function is

$$R(z) = \sum_{k=1}^{4} E_k \frac{a_{0,k} + a_{1,k} z^{-1} + a_{2,k} z^{-2}}{b_{0,k} + b_{1,k} z^{-1} + b_{2,k} z^{-2}} \tag{8}$$

where the coefficients are given as follows:

k	E_k	$a_{0,k}$	$a_{1,k}$	$a_{2,k}$	$b_{0,k}$	$b_{1,k}$	$b_{2,k}$
1	0.88997	2.96642	-1.64489	2.96642	2.92188	-1.78036	2.37549
2	0.30826	3.06432	-1.44908	3.06832	2.95842	-1.39770	2.72160
3	0.03230	11.14112	14.70450	11.14111	3.23626	-3.19192	1.14954
4	0.43067	3.59628	-0.38516	3.59628	3.05145	-2.32297	2.20331

The transfer function of the required 2-D lowpass filter is

$$H(z_1, z_2) = R(z_1)\, R(z_2) \tag{9}$$

The amplitude response of this filter is on view in Fig.3. Since the other characteristics, viz. highpass, bandpass and bandelimination are also separable, and have constant response in the pass and stop bands, their realization procedure with equiripple 2-D recursive filters can similarly be established.

```
 π  -33333333333311111111111111
     33333333333311111111111111
     33333333333311111111111111
     33333333333311111111111111
     22222222222211111111111111
     22222222222211111111111111
     11111111111111111111111111
 |   11111111111111111111111111
     22222222222211111111111111
 W2  22222222222211111111111111
 |   33333333333311111111111111
     33333333333311111111111111
 π  -22222222222211111111111111
 -   11111111111111111111111111
 2   33333333333311111111111111
     PQSTUTQQTQP312332211223333
     QRSUVTRQURQ312332211223333
     TUWYYXUUYUT312332211223333
     QQSUUTQQUQQ312332211223333
     QRSUVTRQURQ312332211223333
     TTVXXWTTXTT312332211223333
     UVWYZXVUYVU313332211223333
     PUWYYXUUYUT312332211223333
     SSUWWVSSWSS312332211223333
     QRSUVTRQURQ312332211223333
 0  -PQSTUTQQTQP312332211223333
     '           '            '
     0           π            π
                 -
                 2
                -W1-
```

Fig.3 Amplitude response of the lowpass filter: P=0.9, Q=0.91, ..., Y=0.99, Z=1; and the symbols 1-3 stand for 0.0, 0.005 and 0.01 respectively.

```
 π  -332123223QURPQSTUUTTSRQQPP
     332123223QURPQSTUUUTSRQQPP
     332123223QURQRSUUUUTSRRQQQ
     332123223RVSQRTUVVVUTSRRQQ
     332123223RWTRSUVWWVUUTSRRR
     332123223SXUSTUWXXWVUUTSSS
     332123223TYVTUVXYYXWVUUTTT
     332123223UYVTUWTYYYXWVVUUT
 |   332123223UZWUVXYZZYYXWVUUU
 W2  332123223UZWUVXYZZYYXWVUUU
 |   332123223UYVTUWXYYYXWVUUTT
     332123223SXUSTUWXXWVUUTSSS
 π   332123223QVSQRTUVVUUTSRRQQ
 -  -332123223QURPQSTUUTTSRQQPP
 2   332123223RWTRSUVWWVVUTSRRR
     332123223UZWUVXYZZYYXWVUUU
     332123223QURQQSUUUUPSRRQQQ
     11111111133333333333333333
     11111111122222222222222222
     11111111122222222222222222
     11111111133333333333333333
     11111111122222222222222222
     11111111111111111111111111
     11111111122222222222222222
     11111111133333333333333333
 0  -11111111133333333333333333
     '           '            '
     0           π            π
                 -
                 2
                -W1-
```

Fig.4 Amplitude response of the synthesized highpass filter: P=0.9, Q=0.91, ..., Y=0.99, Z=1; and symbols 1-3 stand for 0.0, 0.005 and 0.01 respectively.

REALIZATION OF STANDARD CHARACTERISTICS USING FREQUENCY TRANSFORMATIONS

We have seen above that the synthesis of lowpass, highpass, bandpass and bandelimination filters to give equiripple amplitude response in the pass and stop bands is equivalent to synthesizing two corresponding 1-D equiripple filters. But, since the synthesis of 1-D lowpass, highpass, bandpass and bandelimination filters can be done with the use of available 1-D lowpass-lowpass, lowpass-highpass, lowpass-bandpass, and lowpass-band elimination frequency transformations, we can make full use of these 1-D frequency transformations in the synthesis of equiripple 2-D filters to realize the 'standard' amplitude responses.

We will now illustrate by two examples the use of 1-D frequency transformations in the synthesis of highpass and bandpass filters. These frequency transformations have been discussed by Constantinides [12].

Example 2: Synthesize a 2-D highpass filter with equiripple amplitude response to meet the following specifications:
Cut-off frequencies along the w_1-axis and w_2-axis = $\pi/3$ radians/sample,
Transition ratio $\geq$ 0.87, Passband ripple ≤ 0.0975, Stopband ripple ≤ 0.01.

Solution: By Theorem 1, this requires the synthesis of two equiripple 1-D highpass filters each, assuming these to be identical, having the following specifications:
Cut-off frequency = $\pi/3$ radians/sample, Passband ripple ≤ 0.05, Stopband ripple ≤ 0.01. Transition frequency, w_T comes to be approximately=0.87. Such a filter can be obtained by applying 1-D lowpass-highpass transformation

$$g(z) = -\frac{z^{-1}+\alpha}{1+\alpha z^{-1}}$$

on a 1-D prototype lowpass filter. The corresponding frequency function is $-k \cot(w/2)$, where k in our case, assuming the cut-off frequency of the prototype 1-D lowpass filter $= \pi/2$ radians/sample, as calculated from the relation $\tan(\pi/2) = k \cot(\pi/6)$ is $1/\sqrt{3}$. And $\alpha = k-1/k + 1 = -0.268$. Therefore, the requisite lowpass-highpass frequency transformation is

$$g(z) = -\frac{z^{-1} - 0.268}{1 - 0.268\, z^{-1}} .$$

The requisite transition frequency, w_{pi}, of the prototype 1-D lowpass filter as calculated from the relation, $\tan(w_{pi}/2) = k \tan(0.23\pi/2)$ is 1.962 radians/sample, and, therefore, the transition ratio of the prototype filter be greater than or equal to 0.8. The prototype filter designed with these specifications as the transfer function R(z) given below:

$$R(z) = \sum_{k=1}^{4} E_k \frac{a_{0,k} + a_{1,k} z^{-1} + a_{2,k} z^{-2}}{b_{0,k} + b_{1,k} z^{-1} + b_{2,k} z^{-2}} \tag{10}$$

where the coefficients are given below:

k	E_k	$a_{0,k}$	$a_{1,k}$	$a_{2,k}$	$b_{0,k}$	$b_{1,k}$	$b_{2,k}$
1	0.88997	2.07199	0.14396	2.07199	2.02111	0.00850	1.98740
2	0.80826	2.16989	0.33977	2.16989	2.03161	-0.10885	1.85955
3	0.03228	10.24668	16.49336	1.24668	2.05652	-1.40306	0.54030
4	0.43067	2.70185	1.40370	2.70185	2.04105	-0.53411	1.42484

Then the required 2-D highpass filter is obtained as

$$H(z_1, z_2) = R(z_1)R(z_2) \left| \begin{array}{l} z_1^{-1} = g(z_1) \\ z_2^{-1} = g(z_2) \end{array} \right.$$

Amplitude response of this filter is shown in Fig.4.

CONCLUSIONS

We have considered the problem of synthesizing 2-D recursive digital filters to realize a specified amplitude characteristic in the (w_1, w_2) plane. It was pointed out that the problem of realizing an arbitrary but conformable amplitude response by a 2-D recursive digital filter presents two basic hurdles. One is due to non-availability of a suitable algorithm for obtaining an approximation of the prescribed amplitude-squared response with a generalized rational function of two variables while satisfying a given error criterion; and the other is due to lack of an algorithm for extracting a stable filter from it. However, using the equiripple property of the product of two equiripple functions, it was shown that the lowpass, highpass, bandpass and bandelimination amplitude characteristics can be realized with equiripple approximation by 2-D recursive filters in the separable form. This result is of great import from the design point of view as it considerably simplifies the synthesis procedure by permitting break up of the problem of designing an equiripple 2-D filter as such directly into that of designing two appropriate w_1-axis and w_2-axis 1-D digital filters with equiripple amplitude responses in the respective pass and stopbands - the design procedures for which are well available now. Our definition of the transition ratio for 2-D filters in terms of the passband and transition band areas could be used to advantage for developing its simple relation between the transition ratios of the individual 1-D filters. This alongwith the relations for passband and stopband ripples is of consequence for the design of appropriate w_1-axis and w_2-axis 1-D filters so that they together can match specified design parameters for the 2-D filter.

Bearing on the above results, it was further seen that the existing 1-D frequency transformations can as well be used for designing 2-D equiripple lowpass, highpass, bandpass and bandelimination filters from a prototype equiripple 2-D lowpass filter.

Two design examples were taken to illustrate and bear out the developed procedures.

ACKNOWLEDGEMENT

The authors are thankful to Dr. R.K. Varshney, Professor & Head, Department of Electrical Engineering, PAU, Ludhiana (India) for providing the typing and mailing facilities.

REFERENCES

1. T. S. Huang, et al., "Image Processing", Proc. IEEE, Vol. 59, No. 11, ~~pp~~ pp 1586-1609.
2. E. G. Zurfluch, "Applications of two-dimensional Linear Wavelength Filtering", Geophysics, Vol. 32, No. 6, pp. 1015-1035.

3. R.N.Brocewell, "Two-dimensional Aerial Smoothing in Radio Astronomy", Aust.J.Physics, Vol.9,No.3,pp 297-314, Sept.1956.

4. E.J.Burr,"Sharpening of Observational Data in two-dimensions", Aust.J.Physics,Vol.8,No.1,pp.30-53, March,1955.

5. A.Mesco , "Gravity interpretation and information theory III. The method of second derivatives",Annales Univ.Sc.Budapestinensis,XI, pp.37-60, 1967.

6. E.L.Hall,"A Comparison of Computations for Spatial Frequency Filtering" Proc.IEEE,Vol.60, No.7, pp.887-891.

7. W.D.Dryer and G.E.Richmond, "Two-dimensional spatial filtering and computers", Proc.Natl.Electronics Conf.Vol.18,pp.529-535,Oct.1962.

8. S.Treital and J.L.Shanks, "Some aspects of fan filtering", Geophysics, Vol.32, No.5, pp.789-800.

9. J.L.Shanks, "The design of stable two-dimensional recursive digital filter", in Proc.Univ.Missouri, Rolla-Mervin, J.Kelly Communications Conf., Rolla, M.O., pp-15-2-1-15-2-6, Oct.1970.

10. S.Treitel and J.L.Shanks, "The design of multistage separable planar filters", IEEE Trans. Geoscience Electronics, Vol.GE-9, No. 1, pp.10-27, Jan.1971.

11 L.B.Hunt, "Numerical smoothing and filtering in n-dimensions", The Computer Journal, Vol. 15, No.1, pp.58-65, Feb.1972.

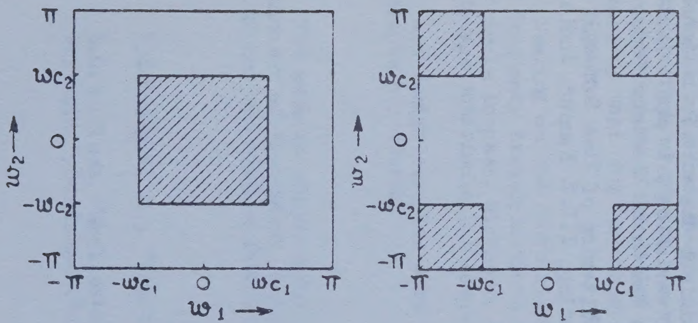
12 A.G.Constantinides, Synthesis of Recursive Digital Filters from Prescribed Amplitude Characteristics. Ph.D.Thesis, University of London, 1968.

13 E.I.Jury, Theory and Application of thez-Transform Method. New York: Wiley, 1964, pp. 73-76.

14 H.S.Sekhon, Analysis and Synthesis of Two-Dimensional Recursive Digital Filters. Ph.D.Thesis, Department of Electrical Engineering, I.I.T. Kanpur, India, Nov.1972.

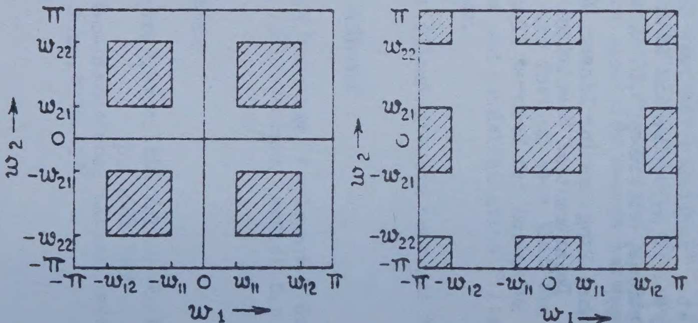
15 L.Collatz, "Inclusion theorems for the minimal distance in rational Tschebyscheff approximation with several variables", in Approximation of Functions, edited by H.L. Garabedian, New York: Elsvier, 1965, pp.43-56.

16 V.P.Sinha, Programs for Linear Digital Filtering of Time Series. Report, Department of Electrical Engineering, I.I.T. Kanpur, India 1968.


π
ωc2
ω2 →
0
-ωc2
-π
-π -ωc1 0 ωc1 π
ω1 →

(a) LOWPASS

(b) HIGHPASS


π
ω22
ω21
ω2 →
0
-ω21
-ω22
-π
-π -ω12 -ω11 0 ω11 ω12 π
ω1 →

(C) BAND-PASS

(d) BAND-ELIMINATION

▨ PASSBAND ☐ STOPBAND

FIG.1. IDEAL AMPLITUDE CHARACTERISTICS OF 2-D FILTERS

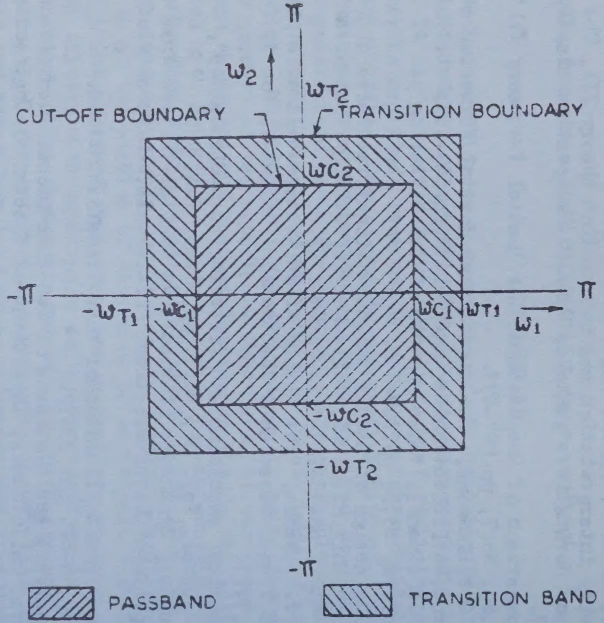

π
ω2 ↑
ωT2
CUT-OFF BOUNDARY
TRANSITION BOUNDARY
ωc2
-π
π
-ωT1
-ωc1
ωc1
ωT1
ω1 →
-ωc2
-ωT2
-π
PASSBAND
TRANSITION BAND

FIG. 2. ILLUSTRATING CUT-OFF AND TRANSITION BOUNDARIES FOR THE IDEALIZED CASE OF A 2-D LOWPASS DIGITAL FILTER

A MULTIRATE DIGITAL FILTER DESIGN METHOD USING CHEBSYSHEV APPROXIMATION

ARTHUR R. BUTZ
Northwestern University
Evanston, Illinois 60201

ABSTRACT

A frequency domain design procedure for determining coefficients of linear or nonlinear phase low pass digital filters is presented. One takes the input sampling frequency $f_s = 2^I f_a$, where f_a is the upper edge of the passband and $I \geq 2$ an integer. The initial cascaded filter stages require no multiplications and the sample rate of their output, which is of course the sample rate of the input to the final stage, is $2f_a$. Coefficients of the final stage are determined by linear and rational Chebyshev approximation. Since the approximation is determined on a simple interval $[0, f_a]$ and, moreover, the function approximated is relatively "nice", various difficulties are avoided and the second Remez algorithm, in a form which does not replace the frequency axis with a discrete grid, may be used to determine approximations. The result is a computationally economical filter, determined by an economical procedure. Group delay, however, is relatively large so the presumption is that design specifications are liberal in that respect.

The methods are easily adapted for the design of bandpass filters when the center frequency of the passband is a composite integer multiple of the passband width.

I. INTRODUCTION

This paper describes a frequency domain design method for digital low pass filters which unites into one approach two recent trends in research on digital filters: multirate digital filters and design via linear or rational Chebyshev approximation. In order for the method to apply, the filter requirements must be such that:

(i) The passband is the interval $[0, f_a]$ and the stopband is the interval $[f_b, f_s/2]$, where $0 < f_a < f_b \leq 2f_a$, f_s is the sampling frequency and $f_s = 2^I f_a$, $I \geq 2$ an integer.
(ii) The passband ripple specification is a constant ± R dB and the stopband attenuation specification is a constant S dB.
(iii) The phase characteristic is either unimportant or must be linear in the passband.
(iv) Design constraints are liberal with respect to group delay.

The method employs the theory of linear and rational Chebyshev approximation to determine filter coefficients of either FIR (finite impulse response) or IIR (infinite impulse response) filters, but the Chebyshev theory is applied only to the passband, the stopband specificiations being satisfied by comb-type filters with unity coefficients. The maximum advantage is obtained when the application is such that it is permissible that the filtered output be generated at rate $2f_a$, but some advantage would also often be obtained when such is not permissible.

The method can be generalized to allow more flexibility in choice of f_s, and is easily extended to the design of bandpass filters whose passband center frequencies are composite integer multiples of passband widths.

II. EXAMPLE: PRELIMINARIES

The design procedure is most effectively elucidated with an example. Suppose a baseband signal $s(t)$, restricted to the baseband $[0,4]$ hz, is corrupted by additive white noise $n(t)$. It is wished to specify a digital (except for analog prefiltering) filter satisfying the specifications that the ripple be within ± .2 dB in the passband [0,4] hz and that the stopband attenuation be at least 50 dB in the stopband (7 hz and above). It is permitted that the filtered output be generated at the rate of 8 hz. Thus in terms of the notation of the previous section $f_a = 4$ hz, $f_b = 7$ hz, $R = .2$ dB, $S = 50$ dB.

Assume that an analog prefilter with transfer function $1/(1 + jf/f_o)$, with $f_o = 26$ hz, is decided on. According to the requirements of our design method the sampling frequency must be chosen from the set 16 hz, 32 hz, 64 hz, etc.. Assume that in order that a suitably low amount of noise be aliased into the passband, it is necessary to sample the output of the prefilter at sampling frequency $f_s = 256$ hz. Thus $f_s = 2^I f_a$, where $I = 6$, and the stopband is $[f_b, f_s/2] = [7,128]$ hz.

Various versions of multirate digital filters have been proposed in recent years [1], [9], [12]. The general idea in such filters is that sample rates are reduced by integer factors at various points inside the filter through the discarding of samples, with the result that the sample rate of the input sequence is some integer multiple of the sample rate of the output sequence. A very specific form of such a multirate filter is proposed here.

The schematic symbol for a factor of two decimator is shown in Fig. 1. The factor of two decimator merely halves a sample rate associated with its input sequence $\{x_i\}$ by discarding every second sample to form a new sequence designated $\{y_k\}$, where $y_k = x_{2k}$.

The complete filter will have the structure shown in Fig. 2. It is seen that there are five sections (referred to here as the comb sections) each having a transfer function of the form $(1 + z^{-1})^{m_i}$, $i = 1,2,3,4,5$ and operating at different sample rates (it is thus a possible abuse of notation to use the same delay operator z^{-1} for all). Multiplications take place only in the final section operating at 8 hz and realizing the transfer function $G(z)$.

It is clear that the overall amplitude function of the prefilter and the comb sections (i.e. excluding the effect of the final stage G), will be of the sort shown in Fig. 3. In order that the ripple specification be met, it is necessary that G compensate for the attenuation in the interval [0,4] hz. Thus, the amplitude function of G, since it is even and periodic with period 8 hz, must be as shown in Fig. 4, and the functions of Figs. 3 and 4, when added, must satisfy the ripple and attenuation specifications.

Once the m_i, $i = 1,2,3,4,5$, have been determined, the values of the function of Fig. 3 in the interval [0,4] hz are determined and thus the ideal amplitude function of G in that interval is determined as the reciprocal (or, in terms of dB, the negative), and thus the ideal $|G|$ is determined for [0,128].

The m_i are determined such that, when the function of Fig. 4 is added to the function of Fig. 3, the stopband attenuation specifications are met. Since the ripple R, in dB, is always small compared to the attenuation S, in dB, it is satisfactory, in the process of determining the m_i, to imagine that G is ideal so that there is 0 ripple in the passband and that the amplitude function of G in the passband is merely the reciprocal of the amplitude function of the rest of the filter in the passband.

A feature of the present design method is the designation of specific frequencies at which one must examine the amplitude function in the process of determining the m_i. If $I = 2$ it is trivially obvious that the relevant set of frequencies is $[f_b, 2f_a]$. If $I \geq 3$ one examines the amplitude function at the set of critical frequencies $(2^{I-1} - 1)f_a$, $(2^{I-2} - 1)f_a, \ldots, 7f_a$, $3f_a$, and also in the intervals $[f_b, 2f_a) \cup (4f_a - f_b, 3f_a)$. The justification for this is given in the Appendix. At present, we proceed with the design problem, wherein $f_a = 4$ hz and $I = 6$.

The first critical frequency is $f = 124$ and the study of the amplitude function at that frequency determines m_1. The amplitude function of the prefilter and the filter section operating at 256 hz will be of the sort shown in Fig. 5. The amplitude function is down A dB at 4 hz and B dB at 124 hz. Since the part of the filter remaining to be designed (the four other sections with transfer functions of the form $(1 + z^{-1})^{m_i}$ plus ideal G) must compensate for the attenuation of A dB at 4 hz, and since that remaining part has an amplitude function even about the center frequency 64 hz, the compensation of A dB at 4 hz will also add A dB to the amplitude function at 124 hz. Thus m_1 must be chosen so that

$$B - A \geq S = 50 \text{ dB}. \tag{1}$$

Since an increase of m_1 has a stronger effect, in terms of dB, at 124 hz than at 4 hz, it is known that a value of m_1 satisfying (1) can be found. In the present example, it is found that $m_1 = 2$ is required.

The remaining m_i, save the last, are determined in order via the same logic. To determine m_2, one examines the amplitude function associated with the prefilter, 256 hz section (with $m_1 = 2$) and the 128 hz section, defines A and B in terms of the amplitude function at 4 hz and 60 hz respectively, and chooses m_2 high enough so that (1) is satisfied. Proceeding in this manner one determines $m_2 = 2$, $m_3 = 3$ and $m_4 = 6$ by successively examining the amplitude function at the critical frequencies 60 hz, 28 hz, 12 hz.

The determination of m_5 is only slightly more complicated, since there is not a single critical frequency involved but the set $[7,8) \cup (9,12)$; m_5 must be large enough so that (1), with A and B suitably defined, holds everywhere in this set. It is found that $m_5 = 3$ is required. A simple computer program determines the m_i in negligible time.

The process just described determines the m_i by requiring them to be such that certain obviously necessary conditions are satisfied. The reasons why these conditions may also, practically speaking, be considered sufficient are explained in the Appendix.

With these values of the m_i, the amplitude fucntion of the digital filter would be as in Fig. 6 if ideal compensation in the passband were provided by G. That ideal amplitude function $|G|$ is

$$H(f) \triangleq 2^8 \sqrt{\frac{1 + (f/f_o)^2}{\prod_{i=1}^{5} [1 + \cos(2^i \pi f/f_s)]^{m_i}}} \tag{2}$$

where $0 \leq f \leq 4$ hz, $f_o = 26$ hz, $f_s = 256$ hz. The factor 2^8 is for the purpose of normalizing $H(0)$. The function $H(f)$ is plotted in Fig. 7 on an absolute (not dB) scale.

G must be chosen so that $|G| \approx H$ in the passband. Note that the part of the filter preceding G has a linear phase characteristic in the band $[0,8]$ hz (except for the small deviation from linear phase caused by the prefilter). Thus if G has linear phase in the passband, the complete filter has linear phase in the passband.

It is well known that a linear phase filter can be derived by approximating a linear phase transfer function (excluding a complex multiplier of unity magnitude and linear phase) with the form

$$R_{L,0}(f) \triangleq \sum_{i=0}^{L} p_i \cos^i(\omega h) \tag{3}$$

where $\omega = 2\pi f$. The result is a linear phase filter of length $2L + 1$. It is also well known that a general nonlinear phase transfer function can be derived by approximating the desired squared amplitude function by

$$R_{L,M}(f) \triangleq \frac{\sum_{i=0}^{L} p_i \cos^i(\omega h)}{1 + \sum_{i=1}^{M} q_i \cos^i(\omega h)} \tag{4}$$

The result is a filter with L zeroes and M poles, so unless $M = 0$ it is an IIR filter. At this point the design problem of the example is set aside and is returned to below.

III. CHEBYSHEV APPROXIMATION

The linear and rational Chebyshev problems are to determine the p_i and q_i in (3) and (4), respectively (however the theory of course allows functions more general than cosines) so as to minimize the maximum absolute value of

$$\delta_{L,M}(f) = \frac{R_{L,M}(f) - \mu(f)}{\nu(f)} \tag{5}$$

on the closed subset F of the finite interval $[a,b]$, where μ and ν are continuous on $[a,b]$ and $\nu > 0$.

There has been great success [6], [7], [10], [11] in using linear Chebyshev approximation with forms (3) to determine coefficients of conventional linear phase filters. The set F is the union of stopband and passband, or $F = [0,f_a] \cup [f_b,f_s/2]$, and $1/h = f_s$. There are few difficulties since it is known that a unique solution exists and the very efficient second Remez algorithm which converges unconditionally and quadratically under the relevant conditions, may be used ([8],pp.16,111). The set F is replaced by a discrete grid, although it is not clear that such discretization is necessary.

There has also been progress [2], [5] using rational Chebyshev approximation of squared amplitude functions with forms (4) to determine coefficients of conventional nonlinear phase filters. Again the set $F = [0,f_a] \cup [f_b,f_s/2]$ and $1/h = f_s$. There are various difficulties. With such a definition of the set F the solution does not necessarily exist, and its existence neither implies nor is implied by the existence of a solution to the discretized version of the problem ([8],p.133). If a unique solution of the rational Chebyshev problem exists, the Remez algorithm is convergent only if one makes a "sufficiently good" guess of the extrema f_i of the error curve [3]. Since these critical f_i strongly tend to be very irregular when $M > 0$, the Remez algorithm essentially cannot handle the task when $F = [0,f_a] \cup [f_b,f_s/2]$, and successful approaches have employed more costly alternatives to the Remez algorithm which require the discretization of the set F and the solution of a minimization problem in each iteration of the unconditionally convergent (quadratically, in [5]) algorithms.

The principle features of the digital filter structure proposed here, in relation to the problem of applying rational Chebyshev approximation to determine coefficients, is that the set F becomes a simple interval, the passband $[0,f_a]$, and the ideal or desired function $\mu(f)$ (i.e. $H(f)$ or $H^2(f)$ in our design example) is relatively "nice" as compared to a desired function embodying the transition from passband to stopband. The implications of this are weightiest in the case of IIR filters. Since F is a simple interval, the Chebyshev problem possesses a unique solution ([8],pp.133,158).

Since the function is nice, it is reasonable to anticipate success in making sufficiently good guesses of the critical f_i. The implications in the case of FIR linear phase filters seem to revolve only around the fact that the order of the linear Chebyshev approximation sought, in the present method, is vastly reduced.

In addition to advantages gained purely in relation to the problem of obtaining Chebyshev solutions, the present method also offers advantages purely on account of the digital filter structure proposed, assuming all coefficients have been determined via Chebyshev approximation.

When approximating $H(f)$ of (2) (FIR linear phase filter) we take $\mu(f) = H(f)$. Since a constant ripple bound is specified, we also take $\nu(f) = H(f)$. When approximating $H^2(f)$, we take $\mu(f) = \nu(f) = H^2(f)$. In both cases the set F on which the approximation is determined is the passband $[0,f_a]$, and $1/h = 2f_a$.

The specific version of the second Remez algorithm that was employed here is that of Cody, et al. [4], which does not discretize the problem. Since the present design method has certain special features, the procedure surmis given in [4] was not used to guess the locations of $L + M + 2$ error extrema. For functions (and their squares) such as shown in Fig. 7, it should be expected that the error extrema either be approximately uniformly spaced or tend to be more dense near the upper edge of the passband, especially when one is dealing with $H^2(f)$ rather than $H(f)$. Therefore, surmis was rewritten so that, for any given problem, a sequence of initial guesses of the f_i were provided, with the $k + 1$ th in the sequence being resorted to only if divergence were obtained using the k th set of initial guesses. Specifically, let $N = L + M + 1$. The first guess ($k = 1$) of the location of $N + 1$ extrema was that they were uniformly spaced and subsequent guesses ($k = 2,3,\ldots$) were such that the extrema were progressively more dense near f_a:

$$f_i = f_a[1 - (1 - i/N)^k] \qquad i = 0,1,\ldots,N \tag{6}$$

The machine used was a CDC 6400, and programs were written in RUN FORTRAN.

IV. EXAMPLE: DESIGN OF FILTER.

In deriving a linear phase filter by approximating $H(f)$ with the form (3), the program was run with progressively higher values of L until the ripple specification was met. In all such cases convergence was obtained with the first initial guess of the $L + 2$ critical points ($k = 1$ in (6)). It was found that $L = 16$ was necessary to meet the ripple specification. The resulting overall amplitude function is essentially as in Fig. 6. The running time of the computer when used for determining the linear Chebyshev approximation with $L = 16$ was about 3 seconds, and correspondingly lower for lower values of L.

A conventional linear phase filter operating at sampling frequency 256 hz and satisfying the present specifications requires a length of 183. The coefficients were determined using the McClellan-Parks program [7]. The computer running times with the various filter lengths that had to be tried in order to determine the minimum length of 183 ranged from about 28 seconds to 50 seconds. No prefilter was involved, but that makes little or no difference in terms of required filter length. In order that a fair comparison be made, this conventional FIR linear phase filter should be imagined as calculating its outputs at rate 8 hz. On this basis the filter requires 1,432 additions/sec and 720 multiplications/sec. By contrast our linear phase filter requires 1,456 additions/sec and 136 multiplications/sec. This comparison assumes, of course, direct implementation via shift registers in both cases and no FFT techniques. It should be noted that FFT techniques would be more profitable as applied to the conventional filter.

It should also be noted that the group delay of the present filter is a liberal 569 input sample intervals, as opposed to 91 for the conventional filter.

Our same program was used to approximate $H^2(f)$ in order to determine coefficients of a nonlinear phase filter. Poles and zeroes outside the unit circle were discarded in order to arrive at a minimum phase filter. All fourteen cases where $1 \leq L + M \leq 4$ were tried, and the program determined a solution in all cases save one. For the four cases where $M = 0$ the program converged with the first guess of $L + 2$ uniformly spaced critical frequencies ($k = 1$ in (6)). In seven of the nine other successful cases the solution was obtained with $k = 1$ in (6), and in these seven cases the ripple specification was also not met. In the two remaining of the nine cases, the solution was obtained with $k = 4$ in (6) and the ripple specification was met. These two cases were $L = 1$, $M = 3$, where the ripple was ± .09 dB, and $L = M = 2$, where the ripple was ± .088 dB. For the case $L = M = 2$ $G(z)$ has two zeroes at $z = -.876009$ and $z = .0437843$ and two poles at $z = -.898191$ and $z = -.564845$. For the case $L = 1$, $M = 2$ the algorithm diverged for $k = 1,2,3,4,5$ in (6). All of the results concerning the fourteen values of L and M, as described in the preceding, were obtained in one computer run with a total running time of about 12 seconds. It is unfortunate that the algorithm diverged for $L = 1$, $M = 2$, because for various reasons it is believed that the solution in that case satisfies the ripple specification.

A comparison of the present IIR filter with a conventional nonlinear phase filter operating at a single sampling frequency of 256 hz is not straightforward. The lowest order (i.e., number of poles plus number of zeroes) such filter seems to require 2 pairs of complex poles plus a real pole, two pairs of complex zeroes on the unit circle, and a zero at $z = -1$. However, with such a filter the extent to which one can exploit the fact, that the output need be generated at the rate of only 8 hz, is somewhat limited. The filter structure with which the maximum exploitation is possible requires 1,344 additions/sec and 1,304 multiplications/sec. By contrast our IIR filter with $L = M = 2$ requires 1,232 additions/sec and 40 multiplications/sec.

However the lowest order (in terms of number of poles plus number of zeroes) filter is not the most economical from the point of view of computation when the output need only be generated at a low rate. Even the conventional FIR linear phase filter of length 183, discussed above, is less demanding computationally (assuming a reasonable weighting of addition vs. multiplication) than the conventional five pole, five zero filter. It is not known exactly what would be the structure of a nonlinear phase filter, operating at 256 hz, meeting specifications, and minimizing computation. However it is clear that its rate of computation would be greater than with the present IIR filter, since in the former case each pole corresponds to 256 additions/sec and 256 multiplications/sec.

We also tried a few cases of IIR filters for somewhat altered specifications, with comparable results.

It is likely that the more costly algorithm employed by Dudgeon [5] can often find solutions in those cases in which the Remez algorithm of [4] fails. However the great economy of the latter implies that it should be the first to be tried, if only to delineate the stubborn cases that seem worthy of closer attention.

V. REMARKS

While these methods are very natural to the case where the output is generated at rate $2f_a$, they may also be valuable when it is required that the output be generated at rate f_s. In the latter case one determines all coefficients as in the preceding, except that all sections operate at rate

f_s, and z^{-1} in each section is replaced by z^{-1} raised to the appropriate power of two. In such a case z^{-1} is interpreted uniformly throughout the filter as representing a delay of $1/f_s$ seconds. If this technique is applied to our linear phase filter of the example then the improvement, in terms of arithmetic, over the conventional linear phase filter is even more significant than in the case where outputs are generated at rate $2f_a$. If the technique is applied to our nonlinear phase filter, the improvement over the conventional nonlinear phase filter is less significant.

Another strategy that could be employed when it is required that the output be generated at rate f_s is to first generate it at rate $2f_a$, as with our filters of the preceding section, and then increase the rate in stages according to the interpolation methods of Bellanger [1]. This strategy has not, however, been investigated in detail thus far.

These methods can be extended to the design of bandpass filters when the center frequency of the passband is a power of two multiple of the width $2f_a$ of the passband. The simplest case is where the sampling frequency f_s equals the center frequency. In that case the prefilter is chosen so that it passes the frequencies in the neighborhood of f_s, its amplitude function is essentially symmetrical about f_s in the passband and goes to zero monotonically, on either side of the passband, rejecting frequencies in the neighborhood of 0 and $2f_s$. Then the design procedure for low pass filters as presented here may be applied, with a mathematical prefilter amplitude function at frequency f, $0 \le f \le f_s/2$, being defined as the maximum of the real prefilter amplitude function at the frequencies $f_s \pm f$. Although the final stage G is then determined in exactly the same way as above, the recovery of the desired signal in the neighborhood of f_s would normally require that the output be generated at rate $4f_a$, unless special properties of the signal permit rate $2f_a$. It is also possible to make the sampling frequency f_s an integer fraction of the center frequency, but then the real prefilter must have a sharper cutoff.

Although we have not studied in detail the problems associated with finite word length in our filter structure, there does not appear to be any disadvantageous feature and it is suspected that it will usually be possible to use shorter coefficient word lengths in our filter as compared to conventional filters. In regard to errors contributed by round-off and truncation, it is only remarked that our filter structure simplifies the analysis of such effects to some degree, since there are no multiplications in the filter before the last stage.

Virtually the same methods may be applied with sample rate decimators which divide sample rates by factors of three rather than two, with the transfer function of the section preceding a factor of three decimator being $(1 + z^{-1} + z^{-2})^{m_i}$. The input sample rate to G must again be $2f_a$. The principal feature offered by factor of three decimators appears to be greater freedom in choosing the initial sampling frequency. The methods can also be extended to decimation by higher integer factors, with low order recursive realization of comb sections in some cases, but various complications arise. Such higher order decimators offer corresponding freedoms in relationships between passband widths and center frequencies in bandpass filters, where such freedoms are usually more important.

Finally, it is obvious that a non-uniform ripple specification can be handled by letting $\nu(f)$ be other than $\mu(f)$. However in such a case one can say little here about the general locations of ripple extrema. A non-uniform attenuation specification, however, is simple to take into account, as remarked in the Appendix.

APPENDIX

In our filter structure, where $f_s = 2^I f_a$, $I \ge 3$, $0 < f_a < f_b \le 2f_a$, f_a is the upper edge of the passband and f_b is the lower edge of the stopband, it is not difficult to show that, when $|G|$ is ideal and the amplitude function of the prefilter is monotone non-increasing, the maximum of the filter amplitude function in the interval $[f_b, f_s/2]$ is in the subset

$$[f_b,2f_a) \cup (4f_a-f_b,3f_a] \cup (8f_a-f_b,7f_a] \cup \cdots\cdots \cup (2^{I-1}f_a-f_b,(2^{I-1}-1)f_a].$$

This shows that in choosing the m_i so that the attenuation specification will be met with ideal $|G|$, we may restrict our attention to one interval $[f_b,2f_a)$ and I-2 intervals of the width of the transition band, but the design procedure employed here restricted our attention to I-2 critical frequencies $(2^{I-1}-1)f_a$, $(2^{I-2}-1)f_a,\ldots,7f_a,3fa$ (the upper edges of the I-2 bands of the width of the transition band) plus the set $[f_b,2f_a) \cup (4f_a-f_b,3f_a)$. The reasons for this are as follows.

Consider our design example wherein $f_a = 4$ hz, $f_b = 7$ hz, $R = .2$ dB, $S = 50$ dB, $I = 6$ so $f_s = 256$ hz. The first critical frequency is 124 hz, at which the ideal squared amplitude function T(f) is examined in order to choose m_1. Given a trial m_1 we may calculate T(f), assuming ideal $|G|$, only in $[0,4]$ and $[124,128]$. This means that, in regard to the interval $(2^{I-1}f_a-f_b,(2^{I-1}-1)f_a] = (121,124]$, we know T(f) only at the upper edge $f = 124$. Choosing m_1 large enough so that the attenuation specification is met at that point will guarantee that it is met in the interval (121,124] only if the supremum in the interval is attained at f = 124. In this connection three observations can be made. First, there will be a strong tendency for the maximum in the interval $(2^{I-1}f_a-f_b,(2^{I-1}-1)f_a]$ to be at the upper edge. The reason for this is that ideal $|G|$ has a discontinuity in its derivative at $f = (2^{I-1}-1)f_a$. Second, since only integer values of m_1 are involved, it will usually be the case that the attenuation specification will be oversatisfied at $f = (2^{I-1}-1)f_a$ so that, if a maximum is achieved at some point on the interior of the interval $(2^{I-1}f_a-f_b,(2^{I-1}-1)f_a]$, it will probably satisfy the attenuation specification anyway. Third, if it is found that, after all m_i have been determined, the attenuation specification is violated at some point in the set delineated, it is easy enough to back up and make the necessary increase in m_1.

The same logic applies to the determination of $m_2,m_3,\ldots m_{I-2}$, but not to m_{I-1} since, having already determined $m_1,m_2,\ldots,m_{I-2}$ and assuming a value of m_{I-1}, T(f) may be calculated for all f. Thus one can search the entire set $[f_b,2f_a) \cup (4f_a-f_b,3f_a)$, appropriately discretized, for the maximum, or one can use golden section or Fibonacci search.

In the design examples involved in this study, it was the case that the maxima (after all m_i had been determined and ideal $|G|$ was assumed) in the intervals $[8f_a-f_b,7f_a]$, $[16f_a-f_b,15f_a],\ldots[2^{I-1}f_a-f_b,(2^{I-1}-1)f_a]$ were indeed in each case at the upper edge.

In choosing, at each stage in the determination of the m_i, the minimum value of m_i that satisfies (1), one will normally get a set of m_i such that the computations (number of additions/sec) preliminary to the final stage G are minimized, but this does not appear to be absolutely guaranteed. There may be cases where an increase of m_i over the minimum required value will allow a reduction of some $m_{i'}$, $i' > i$, such that there are fewer computations, but such cases would certainly be peculiar for multirate filters.

It is simple to apply these methods when there is a non-uniform attenuation specification. Suppose that it is required that the squared amplitude function of the digital filter be $\le \alpha(f)$ in $[f_b,f_s/2]$. If one adds the condition that $P(f)/\alpha(f)$ is monotone non-increasing, where P(f) is the squared amplitude function of the prefilter, then the above holds for the function $T(f)/\alpha(f)$ as well as for the function T(f). One may therefore examine the amplitude function at the same critical frequencies in determinimg the m_i, except that in place of (1) one has $B - A \ge -10 \log_{10} \alpha(f)$, where f is the frequency at which the value of B is calculated.

REFERENCES

[1] M. Bellanger et. al., IEEE Intl. Conf. Comm., Seattle, 1973, 23-11.
[2] C. Charalambous, IEEE Trans. Acoustics etc., Aug. 1975, 333.
[3] W.J. Cody, SIAM Rev., July 1970, 400.
[4] W.J. Cody et. al., Numerische Math., 1968, 242.
[5] D.E. Dudgeon, IEEE Trans. Acoustics etc., Dec. 1974, 443.
[6] J.H. McClellan & T.W. Parks, IEEE Trans. Circ. Th., Nov. 1973, 697.
[7] J.H. McClellan et. al., IEEE Trans. Audio etc., Dec. 1973, 506.
[8] G. Meinardus, Approximation of Functions, Springer, Berlin, 1967.
[9] R.A. Meyer & C.S. Burrus, IEEE Trans. Circ. Sys., Mar. 1975, 162.
[10] T.W. Parks & J.H. McClellan, IEEE Trans. Circ. Th., Mar. 1972, 189.
[11] L.R. Rabiner et. al., Proc. IEEE, April 1975, 595.
[12] R.R. Shively, IEEE Trans. Acoustics etc., Aug. 1975, 353.

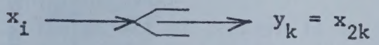


Fig. 1. FACTOR OF TWO DECIMATOR

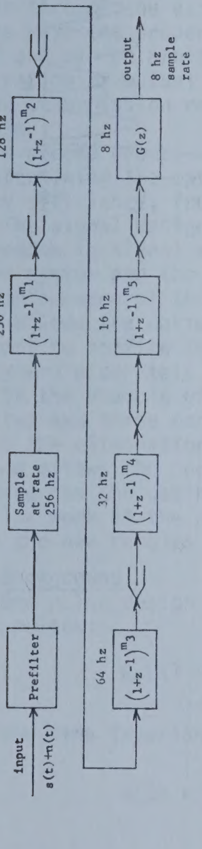

input
s(t)+n(t)
Prefilter
Sample at rate 256 hz
256 hz
$(1+z^{-1})^{m_1}$
128 hz
$(1+z^{-1})^{m_2}$
64 hz
$(1+z^{-1})^{m_3}$
32 hz
$(1+z^{-1})^{m_4}$
16 hz
$(1+z^{-1})^{m_5}$
8 hz
G(z)
output
8 hz sample rate

Fig. 2. FILTER STRUCTURE

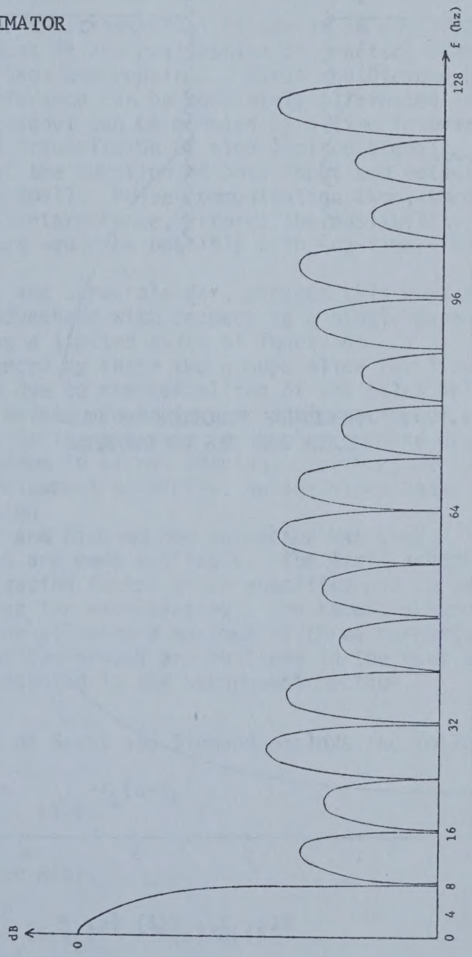


Fig. 3. AMPLITUDE FUNCTION OF COMB SECTIONS

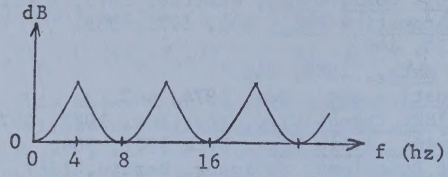


Fig. 4. AMPLITUDE FUNCTION OF FINAL SECTION G

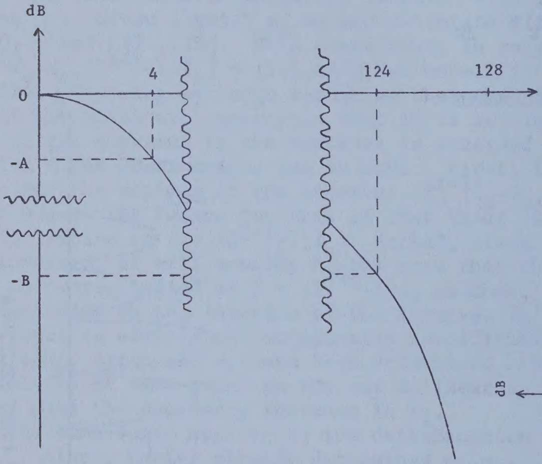


Fig. 5. AMPLITUDE FUNCTION OF PRE-FILTER AND 256 HZ SECTION

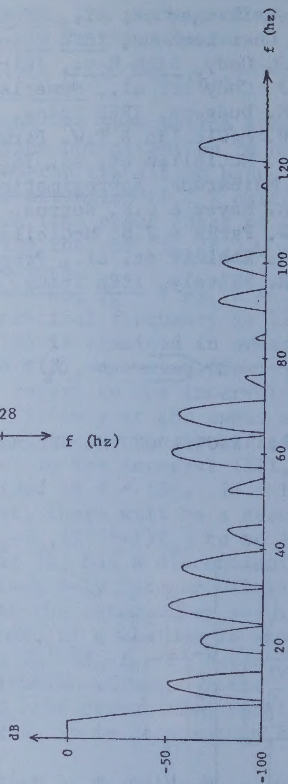


Fig. 6. AMPLITUDE FUNCTION OF OUR FILTER WHEN G PROVIDES IDEAL COMPENSATION

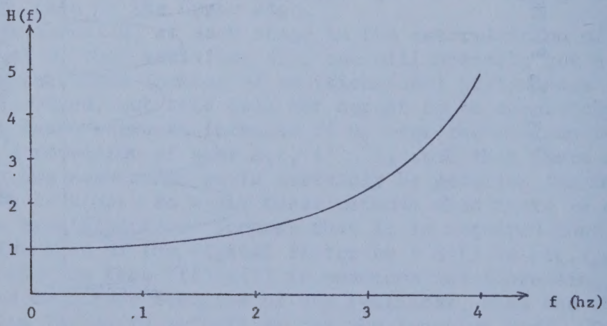


Fig. 7. H(f) vs. f

OPTIMIZATION OF TIME LIMITED OUTPUTS IN BAND LIMITED CHANNELS

PHILIPP DINES and DOV HAZONY
Department of Electrical Engineering and Applied Physics
Case Western Reserve University
Cleveland, Ohio 44106

ABSTRACT

Of concern are time limited output pulses. The poles of the channel are cancelled by zeros of the preshaped input signal. It is shown that it is possible to generate a special family of input pulses which preserves the time limited nature of the output. Optimizing then, yields extremely sharp output pulses. The work is an extention of the results of Gerst and Diamond (1).

1. Introduction

The object of concern is to attain improved signal transmission and thus improved communication through the elimination of intersymbol interference.

In spite of the power of the transversal filter as an equilizer intersymbol interference arises in any realization in practice when a very large or infinite number of taps are required. Gerst and Diamond have shown that intersymbol interference can be completely eliminated for a communication system whose channel can be modeled by a time invariant filter through the exclusive transmission of time limited signals. These pulses have the property that the duration of both input and output are equal and can be arbitrarily small. Pulse communication then, through the elimination of mutual signal interference, affords the possibility of higher transmission rates than would be possible with non-time limited signals.

A second group, Hancock and Schwarzlander, pursued this work further by determining the optimal waveshape with respect to a single parameter, energy efficiency, from among a limited class of functions.

The signal designs advanced by these two groups allow for limited or no freedom in signal shaping due to the cancelling of the poles introduced by the system and the input pulse. To achieve an optimization of the overall performance of the pulse it is necessary to develop a pulse design which allows the maximum freedom in signal shaping. Further, it is of interest to confine this development to digital pulses since these can be easily and accurately generated.

In the example of Gerst and Diamond one parameter was used. In this work two and three parameters are made available. The first scheme involves the elimination of a gating factor which specifies one parameter in two and allows the second free for wave shaping. The second scheme reintroduces the gating factor allowing a maximum of three parameters.

The work of the previous two groups are outlined in the next section while the new results are presented in the subsequent section.

2. Background

One pulse design scheme of Gerst and Diamond employs the following input pulses;

$$E_1(s) = C(s) \prod_{k=1}^{n} (1-e^{-\tau_k(s-\gamma_k)})$$

into the time invariant filter H(s),

$$H(s) = N(s) \prod_{k=1}^{n} (s-\gamma_k)^{-1} = N(s)/D(s) \quad .$$

Here C(s) is a containing function and must be such that the product $E_o(s) = E_i(s)\, H(s)$ is entire. N(s) and D(s) are polynomials with real coefficients, having no common factor, and the order of D(s) being greater than or equal to that of N(s). In addition, the output pulse, $E_o(s)$, must satisfy the bounded conditions specified by Gerst and Diamond.

Applying the above technique to the filter H(s) where

$$H(s) = (\alpha^2 + \beta^2)/[(s + \alpha)^2 + \beta^2] \quad , \tag{2-1}$$

Gerst and Diamond determined the following input-output pair where T is the pulse duration;

$$E_i(s) = \frac{(1-e^{\frac{-Ts}{3}})(1-e^{\frac{-T(s+\alpha-j\beta)}{3}})(1-e^{\frac{-T(s+\alpha+j\beta)}{3}})}{s} \quad , \tag{2-2}$$

$$E_o(s) = \frac{(\alpha^2+\beta^2)(1-e^{\frac{-Ts}{3}})(1-e^{\frac{-T(s+\alpha-j\beta)}{3}})(1-e^{\frac{-T(s+\alpha+j\beta)}{3}})}{s[(s+\alpha)^2+\beta^2]} \quad .$$

Let $u(t-t_o) = 1$ for $(t \geq t_o)$ and zero elsewhere. The following is the time response of the above

$$e_i(t) = u(t)-(1+2e^{\frac{-T\alpha}{3}} \cos \frac{T\beta}{3})u(t-\frac{T}{3})+(e^{\frac{-2T\alpha}{3}} + 2e^{\frac{-T\alpha}{3}} \cos \frac{T\beta}{3})u(t-\frac{2T}{3})$$

$$-e^{\frac{-2T\alpha}{3}} u(t-T) \quad ,$$

$$e_o(t) = r(t)-(1+2e^{\frac{-T\alpha}{3}} \cos \frac{T\beta}{3})r(t-\frac{T}{3})+(e^{\frac{-2T\alpha}{3}} + 2e^{\frac{-T\alpha}{3}} \cos \frac{T\beta}{3})r(t-\frac{2T}{3})$$

$$-e^{\frac{-2T\alpha}{3}} r(t-T) \quad ,$$

where

$$r(t) = (1 - \frac{(\alpha^2+\beta^2)^{\frac{1}{2}}}{\beta} e^{-\alpha t} \cos(\beta t-\tan^{-1} \frac{\alpha}{\beta}))\, u(t) \quad . \tag{2-3}$$

This design scheme leads to easily realizable digital pulses with output pulses having no crossovers. The performance of this scheme at $T=\pi$ with respect to the energy efficiency, η_p , the channel energy capacity transfer efficiency, η_c , and the maximum output to maximum input height ratio, ρ , is exhibited in table one of page 5 . Mathematically, these parameters are given by: $\rho = e_{o_{max}} / e_{i_{max}}$ and

$$\eta_p = \int_0^T e_o^2(t)\, dt \Big/ \int_0^T e_i^2(t)\, dt \quad , \qquad \eta_c = \int_0^T e_o^2(t)\, dt \Big/ e_{o_{max}}^2\, T \quad .$$

Hancock and Schwarzlander optimized with respect to η_p while under no amplitude constraint. They found the optimal signal shape from among a limited class of functions.

For the example of Gerst and Diamond where H(s) is given by (2-3) and $\alpha=\beta=1$, the optimal efficiency is given by

$$1 \,/\, (1 + (3.33/T)^4\,)$$

where T is the duration of the pulse.

3. New Results

The Gerst and Diamond example has one parameter for pulse shaping while that of Hancock and Schwarzlander while optimal with respect to η_p is fixed and is not a pulse.

The original examples reported here are strictly digital pulses with two to three pulse shaping parameters.

The first case involves the elimination of the containing function (gating factor) which specifies one of the two available pulse shaping parameters. Two examples are:

a.) $E_i(s) = N_1(s) / s$ where (3-1)

$$N_1(s) = \frac{\sin(b-a) - e^{-\frac{a}{\beta}(s+\alpha)} \sin b + e^{-\frac{b}{\beta}(s+\alpha)} \sin a}{\sin a} , \quad (3\text{-}2)$$

and

b.) $E_i(s) = N_2(s) / s$ where (3-1)

$$N_2(s) = \cos(b-a) - e^{-\frac{a}{\beta}(s+\alpha)} \cos b - e^{-\frac{b}{\beta}(s+\alpha)} \cos a + e^{-\frac{(b+a)}{\beta}(s+\alpha)} .$$

These input pulses cancel the poles of the channel filter given by (2-1) for all values of a and b.

For example in the case of (3-2) substituting $s = -\alpha \pm j\beta$ into $H_1(s)$ for $\sin a \neq 0$ leads to

$$N_1(-\alpha \pm j\beta) = \frac{\sin(b-a) - e^{-\frac{a}{\beta}(-\alpha \pm j\beta + \alpha)} \sin b + e^{-\frac{b}{\beta}(-\alpha \pm j\beta + \alpha)} \sin a}{\sin a}$$

$$= 0 .$$

Similarly, it is possible to show that for $s = -\alpha \pm j\beta$, $N_2(s)$ is zero for all a and b making these two parameters free for pulse shaping. To eliminate the containing function a parameter may be used to cancel the pole at s=0. For example, in (3-1) $N_1(s=0) = 0$ with $\sin a \neq 0$ implies

$$N_1(0) = \sin(b-a) - e^{-\frac{a\alpha}{\beta}} \sin b + e^{-\frac{b\alpha}{\beta}} \sin a = 0 ,$$

$$F(a) = \frac{\cos a - e^{-\frac{a\alpha}{\beta}}}{\sin a} = \frac{\cos b - e^{-\frac{b\alpha}{\beta}}}{\sin b} = F(b) .$$

Similarly, it can be shown that for $N_2(0)=0$ the parameters must be chosen to satisfy

$$-1 / F(a) = F(b) .$$

One parameter is still free for pulse shaping while the second has been specified to insure that the output signal is time limited, that is, that the frequency domain representation of the output, $E_o(s)=E_i(s)\,H(s)$, is entire.

The time domain representations of $E_i(s)$ and $E_o(s)$ are the following;

$$e_i(t) = \frac{\sin(b-a)}{\sin a} u(t) - \frac{e^{-\frac{a\alpha}{\beta}} \sin b}{\sin a} u(t - \tfrac{a}{\beta}) + e^{-\frac{b\alpha}{\beta}} u(t - \tfrac{b}{\beta}), \quad (3\text{-}3)$$

with the resulting output pulse

$$e_o(t) = \frac{\sin(b-a)}{\sin a} r(t) - \frac{e^{-\frac{a\alpha}{\beta}} \sin b}{\sin a} r(t - \tfrac{a}{\beta}) + e^{-\frac{b\alpha}{\beta}} r(t - \tfrac{b}{\beta}),$$

where r(t) is given by (2-3). The above pair corresponds to the "a" input of (3-1) while the following pair corresponds to the "b" input of (3-1).

$$e_i(t) = \cos(b-a)\,u(t) - e^{-\frac{a\alpha}{\beta}} \cos b\; u(t-\frac{a}{\beta}) - e^{-\frac{b\alpha}{\beta}} \cos a\; u(t-\frac{b}{\beta})$$

$$e^{-\frac{(b+a)\alpha}{\beta}}\; u(t-\frac{b+a}{\beta}) \quad , \tag{3-4}$$

with the resulting output pulse being

$$e_o(t) = \cos(b-a)\,r(t) - e^{-\frac{a\alpha}{\beta}} \cos b\; r(t-\frac{a}{\beta}) - e^{-\frac{b\alpha}{\beta}} \cos a\; r(t-\frac{b}{\beta})$$

$$e^{-\frac{(b+a)\alpha}{\beta}}\; r(t-\frac{b+a}{\beta}) \quad .$$

The input pulse (3-3) for $\alpha=\beta=1$ has a strictly positive characteristic while (3-4) has crossovers. The performance in terms of η_p, η_c, and ρ is about the same.

Alternatively, if the gating factor is reintroduced there are three free pulse shaping parameters made available. An example where two parameters have been set equal to each other is

$$E_i(s) = N_1(s)(1 - e^{-\frac{as}{\beta}}) / s \quad , \tag{3-5}$$

where again the output pulse is $E_o(s) = E_i(s)\,H(s)$.

The time domain representations corresponding to the above pair are;

$$e_i(t) = \frac{\sin(b-a)}{\sin a}\,u(t) - \left(\frac{e^{-\frac{a\alpha}{\beta}}\sin b}{\sin a} + \frac{\sin(b-a)}{\sin a}\right) u(t-\frac{a}{\beta})$$

$$+\, e^{-\frac{b\alpha}{\beta}}\, u(t-\frac{b}{\beta}) + \frac{e^{-\frac{a\alpha}{\beta}}\sin b}{\sin a}\, u(t-\frac{2a}{\beta}) - e^{-\frac{b\alpha}{\beta}}\, u(t-\frac{b+a}{\beta}) \quad ,$$

$$e_o(t) = \frac{\sin(b-a)}{\sin a}\,r(t) - \left(\frac{e^{-\frac{a\alpha}{\beta}}\sin b}{\sin a} + \frac{\sin(b-a)}{\sin a}\right) r(t-\frac{a}{\beta})$$

$$+\, e^{-\frac{b\alpha}{\beta}}\, r(t-\frac{b}{\beta}) + \frac{e^{-\frac{a\alpha}{\beta}}\sin b}{\sin a}\, r(t-\frac{2a}{\beta}) - e^{-\frac{b\alpha}{\beta}}\, r(t-\frac{b+a}{\beta}) \quad , \tag{3-6}$$

where r(t) is given by (2-3).

The performance of this input - output pair for a pulse duration, T, equal to π is compared with that of the previous two groups in the following table for H(s) given by (2-3) for $\alpha=\beta=1$.

Table I

Pulse	ρ	η_c	η_p
(2-2) (G-D)	59%	38%	29%
(H-S)	58%	46%	44%
(3-5)	75%	59%	33%

4. Conclusion

In figure one and two the input and output signals, respectively, that correspond to the data in table one are plotted. In figure one all the input signals have a maximum of unity. The resulting output signals are normalized to the greatest maximum of the three signals compared, that is to the maximum of (3-6).

From figure two it is clear that for a unity maximum input signal the output with the largest peak, greatest ρ, is that of (3-6). This output pulse, (3-6), is very sharp and very nearly symmetric. At the price of more input crossovers the overall performance is greater with the reintroduction of the gating factor.

A simpler pulse is achieved at the price of reduced total performance. For (3-1) of duration 5.0 second the performance parameters are; $\eta_p = 65\%$, $\rho = 86\%$, and $\eta_c = 29\%$. In this example there are no crossovers for the input pulse.

References

1. I. Gerst and J. Diamond, "The Elimination of Intersymbol Interference by Input Signal Shaping", Proc. IRE, vol. 49, pp. 1195-1203, 1961.

2. J.C. Hancock, D.G. Lainiotis, J.C. Lindenlaub, R.G. Marquart, and H. Schwarzlander, "Advanced Communication Theory Techniques", Technical Report No. TR-EE63-2, School of Engineering, Purdue University, pp. 101-103, March 1963.

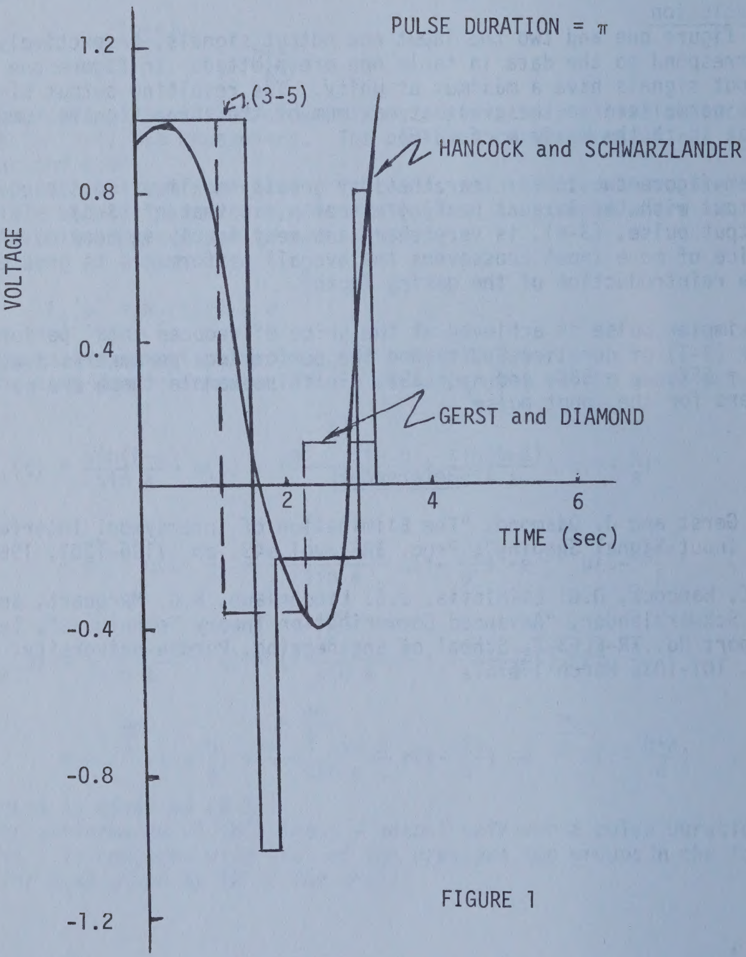

PULSE DURATION = π
(3-5)
HANCOCK and SCHWARZLANDER
GERST and DIAMOND
VOLTAGE
1.2
0.8
0.4
-0.4
-0.8
-1.2
2
4
6
TIME (sec)

FIGURE 1

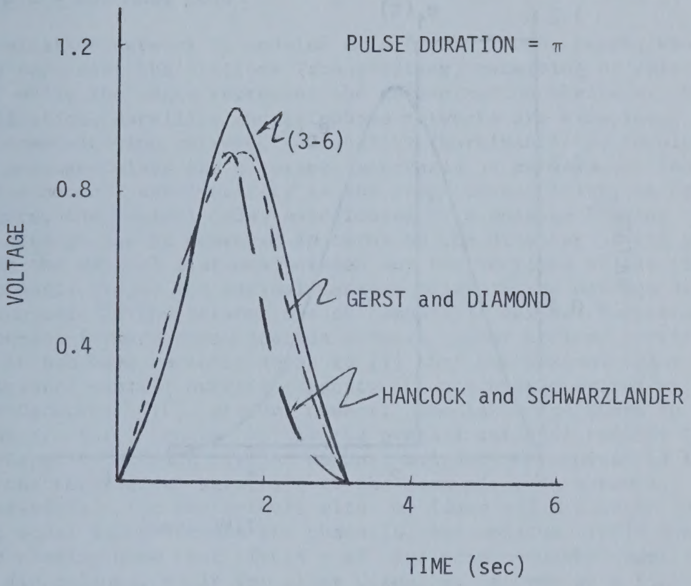

1.2
0.8
0.4
VOLTAGE
PULSE DURATION = π
(3-6)
GERST and DIAMOND
HANCOCK and SCHWARZLANDER
2
4
6
TIME (sec)

FIGURE 2

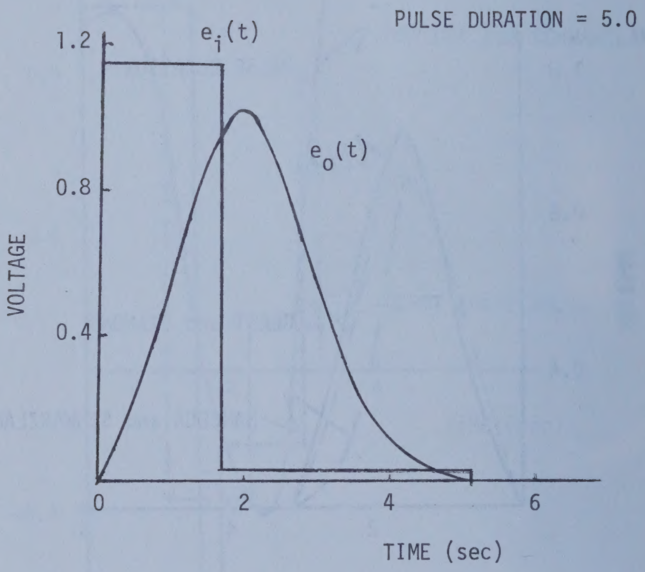


FIGURE 3

COMMUNICATION NETWORKS UNDER DELAY AND CONNECTIVITY CONSTRAINTS*

JEHUDA HARTMAN and IZHAK RUBIN
Department of System Science
School of Engineering and Applied Science
University of California
Los Angeles, California 90024

ABSTRACT

A communication network is modeled as a linear graph, its survivability is measured in terms of the graph connectivity, and its maximal message delay index is expressed in terms of the diameter of the graph. To characterize extremal networks under reliability and delay constraints, a lower bound is obtained on the number of lines of any k-connected graph with diameter d. Graphs achieving the latter lower bound are synthesized. Also obtained are lower and upper bounds on the number of lines of k-connected graphs with diameter d and a prescribed number of vertices.

1. Introduction

A communication network is modeled as a finite linear graph, where the vertices represent the stations (transmitting, receiving or relaying information) while the edges represent the communication channels. Computer-communication, satellite and telephone networks are examples. In designing a communication network, reliability (survivability, invulnerability) and message-delays are of prime importance as performance indices. A measure of a network survivability is its graph connectivity, as defined below. In turn, the maximal delay experienced by a message flowing through the network can be measured in terms of the diameter of the graph (representing the maximal distance between any two vertices of the graph.) The latter clearly yields the maximal message delay if the message delays across the channels in the network are of comparable value. Furthermore, for any store-and-forward communication network, under a fixed routing discipline, it has been recently shown in [1] that the maximal delay γ and the associated minimal overall capacity C are characterized by a unique Delay-Capacity (γC) product number. The latter is shown in [1] to be the sum of $\gamma\lambda_I$, where λ_I is the overall internal traffic flow, and a Delay-Capacity product $(\gamma C)^*$ number, uniquely determined by the routing and the topological structure of the communication network. It is readily observed that for any network with m lines and a diamter d, by assigning equal delays across the channels, one obtains $(\gamma C)^* \leq md$. One can also readily show that $(\gamma C)^* = md$ for many networks under various routing disciplines, while for other cases md serves as a tight bound. Thus, we are interested in studying the characteristics of survivable (k-connected) networks with small md values.

In this paper, considering networks with prescribed connectivity and diameter values, we obtain in Section 3 a lower bound on m (and subsequently md) and show how to synthesize a family of k-connected, diameter d, graphs achieving the latter minimal md value (containing thus the minimal number of lines). Simple bounds on m are then obtained, in Section 4, when considering such graphs with a prescribed number of vertices.

---

*This work was supported by the office of Naval Research under Contract Number N00014-75-C-0609, and by the National Science Foundation under Grant Number ENG 75-03224.

2. Terminology and Notations

All graphs considered in this paper are undirected, without loops and multiple edges. By $V(G)$ and $E(G)$ we denote the set of the vertices and the set of edges, respectively, and we set $v(G) = |V(G)|$, $e(G) = |E(G)|$. A graph G with $v(G) \geq k+1$ is called k-connected if between any pair u,v, of distinct vertices of G there exist k u,v paths in G, any two of which have exactly u,v in common. It is obvious that a k-connected graph cannot be disconnected by removing less than k vertices. The converse was proved by Menger and Whitney and is summarized by the following theorem: "A graph G is k-connected if and only if $v(G) \geq k+1$, and it is impossible to disconnect G by removing $k-1$ or fewer vertices" (see [2]).

Dirac in [6] gave another useful characterization of k-connected graphs. "A graph is k-connected if and only if $v(G) \geq k+1$ and for every vertex u and every $T \subset G$ with $v(T) = k$ and $u \notin T$, there exists a u,T-fan in G (which is a union of k u,T-paths which have only u in common).

The distance $d(u,v)$ between two vertices $u,v \in V(G)$ is the length of the shortest path joining u and v. The diameter of G, $d(G)$, is defined as

$$d(G) = \max_{u,v \in V(G)} d(u,v) .$$

For further definitions the reader is referred to [2].

In [3], graphs with given number of vertices and given edge-connectivity, having minimal number of edges were constructed. The more general problem of determining the minimal number of edges required for k-connectivity on a given number of vertices was solved in [4]. The construction of [4] will be used in this paper.

Let $H(k,d)$ denote the class of k-connected graphs with diameter d. The subclass of graphs in $H(k,d)$ having exactly n vertices is denoted by $H(n,k,d)$. Let $g(k,d)$ and $g(n,k,d)$ denote the minimal number of edges of graphs within $H(k,d)$ and $H(n,k,d)$, respectively. The graphs with minimal number of edges in these classes will be called extremal graphs of the respective classes.

In this paper we get bounds on $g(k,d)$ and $g(n,k,d)$ and find extremal graphs in various cases.

3. Bounds on g(k,d) and the Synthesis of an H(k,d)-extremal graph

It is known [5] that if $G \in H(k,d)$, then

$$v(G) \geq k(d-1)+2 . \tag{1}$$

Further this inequality is best possible. Using (1) and the fact that every vertex of a k-connected graph has degree k or more, we obtain the following lower bound on $g(k,d)$:

$$g(k,d) \geq \frac{v(G)k}{2} \geq \frac{(k(d-1)+2)k}{2} . \tag{2}$$

Theorem 1 provides an upper bound on $g(k,d)$ for even-connectivity.

Theorem 1: For $\ell \geq 1$, $d \geq 1$,

$$g(2\ell,d) \leq (2\ell(d-1)+2)\ell . \tag{3}$$

Proof: Let us label the k vertices of the complete graph of k vertices K_k by v_1, $v_2, \ldots, v_k$. Delete from K_k the edges $(v_i, v_{i+\frac{k}{2}})$, $i = 1,2,\ldots,\frac{k}{2}$ $(k=2\ell)$. Denote the resulting graph by H. H is one of the graphs constructed in [4], and shown to be $(k-2)$-connected. H is also regular of degree $k-2$.

Let H_1, $H_2, \ldots, H_{d-1}$ be $d-1$ disjoint copies of H. The k vertices of H_j will be denoted by v_{1j}, $v_{2j}, \ldots, v_{kj}$, $j=1,2,\ldots,d-1$.

Form G in the following way: Join a new vertex u_1 to each of the vertices of H_1, and a vertex u_2 to each of the vertices of H_{d-1}. Then join v_{ij} to v_{ij+1} by an edge for all $i=1,2,\ldots,k$ and $j=1,2,\ldots,d-2$. The graph with $k=6$ $d=3$ is shown in figure 1. By our construction,

$$e(G) = \frac{(k(d-1)+2)k}{2}.$$

Since any pair of vertices in G is contained in a cycle of length $\leq 2d$, we have $d(G) \leq d$. However, $d(u_1, u_2) = d$ which shows that $d(G) = d$. In order to show that $G \in H(k,d)$ and thus establish (3), the k-connectivity of G must be shown. Let a,b be two distinct vertices of G. If $a=u_1$ and $b=u_2$ we have k vertex disjoint a,b paths in G. Let us now examine the other cases.

Case 1: $a=u_1$ and $b \in H_j$ $(1 \leq j \leq d-1)$.
From symmetry, we may assume $b=v_{1j}$, and let

$$T = H_j - \left\{v_{1j}, v_{1+\frac{k}{2}j}\right\}$$

Since H_j is $(k-2)$-vertex connected we may apply Dirac's theorem to obtain a b,T-fan in H_j. Here, as in the following cases, the existence of such a fan which does not contain $v_{1+\frac{k}{2}j}$ can be deduced from the special form of H_j. By our construction there exist $k-2$ vertex disjoint a, u_{ij} paths G for $1 \leq i \leq k$.

The appropriate paths combined with the b,T-fan yield $k-2$ disjoint a,b paths. The cycle through a,b, u_2, $u_{1+\frac{k}{2}j}$ is disjoint from the previous paths and provides 2 more a,b paths.

Case 2: $a \in H_\ell$, $b \in H_j$ and $\ell=j$.
We have $k-2$ a,b paths in H_j and the cycle through a,b, u_1, u_2, provides 2 additional disjoint paths.

Case 3: $a \in H_\ell$, $b \in H_j$, and $\ell \neq j$.

Let $a=v_{s\ell}$ $b=v_{tj}$ and we have 2 subcases:

(a) $s=t$ (w.l.g. $s=t=1$) .

Take $T_j = H_j - \left\{v_{1j}\ ,\ v_{1+\frac{k}{2}j}\right\}$

$T_\ell \quad H_\ell - \left\{v_{1\ell}\ ,\ v_{1+\frac{k}{2}\ell}\right\}$

Combine a b,T_j - fan in H_j and a a,T_ℓ - fan in H_ℓ with the shortest $v_{ij},v_{i\ell}$ paths $(1 \le i \le k)$ to establish $k-2$ vertex disjoint a,b paths. The cycle through $a,b,\ u_1\ ,\ u_2\ ,\ v_{1+\frac{k}{2}j}\ ,\ v_{1+\frac{k}{2}\ell}$ provides two more paths.

(b) $s \neq t$.

Take $T_j = H_j - \left\{v_{tj}\ ,\ v_{t+\frac{k}{2}j}\right\}$

$T_\ell = H_\ell - \left\{v_{s\ell}\ ,\ v_{s+\frac{k}{2}\ell}\right\}$

Construct an a,T_ℓ - fan in H_ℓ and a b,T_j - fan in H_j , and proceed as before, to obtain k disjoint a,b paths in G .

Q.E.D.

Combining inequalities (2) and (3) we obtain $g(k,d)$ for even k .

Theorem 2: For $\ell \ge 1$, $d \ge 1$, and $k=2\ell$,

$$g(2\ell,d) = \frac{[k(d-1)+2]k}{2} \ . \tag{4}$$

Note that the graph G constructed in theorem 2 is a graph with a minimal number of edges in class $H(2\ell,d)$. Furthermore any $G \in H(k,d)$ which achieves equality in (2) must be k-regular and have $k(d-1)+2$ vertices.

If k is odd, we construct a graph H on k vertices which is $(k-2)$-connected by the methods of [4]. H cannot be $(k-2)$-regular. For H we have

$$e(H) = \frac{k(k-2)+1}{2} \ .$$

We then construct a graph $G \in H(k,d)$ by using the construction given in the proof of theorem 1, with the graph H there replaced by the present H . We use inequality (2) to conclude the following.

Theorem 3: For k odd, $d \ge 1$,

$$\frac{k(k(d-1)+2)+d-1}{2} \ge g(k,d) \ge \frac{(k(d-1)+2)k}{2} \ . \tag{5}$$

The difference between the upper and lower bound is equal to $(d-1)/2$, being practically a small number.

4. Bounds on $g(n,k,d)$

Considering a k-connected graph with n vertices and diameter d, we first derive upper and lower bounds on $g(n,k,d)$.

Theorem 4: For even $k \geq 2$, $d \geq 2$, and $n \geq (d-1)k+2$, we have

$$\frac{k}{2}(2n-kd+k-2) \geq g(n,k,d) \geq \frac{nk}{2} \tag{6}$$

Proof: According to [5] we know that $n \geq (d-1)k+2$ for all graphs in $H(n,k,d)$. The lower bound used in (6) is the standard lower bound on a k-connected graph with n vertices. To obtain the upper bound on $g(n,k,d)$ the following construction is used. We take the extremal graph in $H((d-1)k+2, k,d)$ as in the proof of Theorem 1. Add the $n-(d-1)k-2$ remaining vertices and join each one of them to all k vertices of H_1 (we use the notation of the respective proof).

The resulting graph will be denoted by $\bar{G}$. From symmetry we see that $d(x,\bar{x}) \leq d$ where $x \in V(G)$ $\bar{x} \in V(\bar{G}-G)$. Furthermore $d(x_1,x_2) = 2$ for $x_1,x_2 \in V(\bar{G}-G)$. Hence $d(\bar{G}) = d$ as before. We also observe that between any point of G and $\bar{G}-G$ there exist k disjoint paths, and the same is true for a pair of vertices within $\bar{G}-G$. Therefore, $\bar{G} \in H(n,k,d)$.

Q.E.D.

We note that $G \in H(n,k,d)$ assumes the lower bound in (6) if and only if G is k-regular. Using the previous results we note that

$$g(n_1,k,d) = \frac{n_1 k}{2}$$

where $n_1 = k(d-1)+2$ and k even.

Another case where a k-regular, k-connected graph with diameter d on n vertices can be constructed, (thus will be an extremal graph of $H(n,k,d)$) is indicated in [7] for a different purpose. We construct a hierarchic tree with $\left[\frac{d}{2}+1\right]$ levels of vertices, in which all vertices have degree k except for the lowest level, where all vertices have degree one. Now take k identical hierarchic trees and combine them in their lowest level. The graph obtained is k-regular, of diameter d, and has n_2 vertices where

$$n_2 = \frac{2k}{k-2}\left((k-1)^{[\frac{d}{2}]}-1\right) . \tag{7}$$

By using the definition of connectivity the graph obtained is k-connected. Hence

$$g(n_2,k,d) = \frac{n_2 k}{2} \tag{8}$$

where n_2 is given in (7).

The case $g(n,k,2)$ was solved by J. A. Bondy and U. S. R. Murty in [8]. They proved that if $n > k^3 + \alpha(n) \cdot \alpha(k)+1$ (where $\alpha(t)$ is 0 or 1 according as t is odd or even) then,

$$g(n,k,2) = \left\lceil \frac{(n-1)(k+1)+1}{2} \right\rceil . \tag{9}$$

Extremal graphs of class $H(n,k,2)$ are obtained by taking a $(k-1)$-connected graph on $n-1$ vertices using methods of [4], and adding a new vertex adjacent to all the previous $n-1$ vertices.

Finally we note that the lower bound in (6) can be achieved in many other cases.

5. Conclusions

Extremal communication networks under delay and connectivity constraints have been studied and synthesized. Lower bounds on the minimal number of lines of k-connected networks with a prescribed diameter d have been derived, and a synthesis procedure presented to obtain graphs achieving these bounds. Upper and lower bounds are presented for k-connected networks with a given diameter d and a prescribed number of vertices. Further characteristics of extremal graphs with prescribed diameter and vertex values, under connectivity constraints, are presently under investigation.

REFERENCES

1. I. Rubin, "The Delay-Capacity Product for Store-and-Forward Communication Networks: Tree Networks" to be published in "Applied Mathematics and Optimization", 1975.

2. F. Harary, "Graph Theory", Addison-Wesley, 1969.

3. D. R. Fulkerson, L. S. Shapley, "Minimal K-arc Connected Graphs", Networks, 1 (1971), pp. 91-98.

4. F. Harary, "The Maximum Connectivity of a Graph", Proc. National Academy of Science, 48 (1952), pp. 1142-1146.

5. M. E. Watkins, "A Lower Bound for the Number of Vertices of Graph", American Mathematical Monthly 74, Part 1, Jan.-May, 1967, pg. 297.

6. G. A. Dirac, "Extensions of Menger's Theorem", J. London Mathematical Society 38 (1963), pp. 148-161.

7. H. P. Friedman, "A Design for (d,k) Graphs", IEEE Transactions Electronic Computers (Short notes) Vol. EC-15, pp. 253-254, April, 1966.

8. J. A. Bondy and U. S. R. Murty, "Extremal Graphs of Diameter Two with Prescribed Minimum Degree," Studia Sci. Math. Hung. (1972), pp. 239-241.

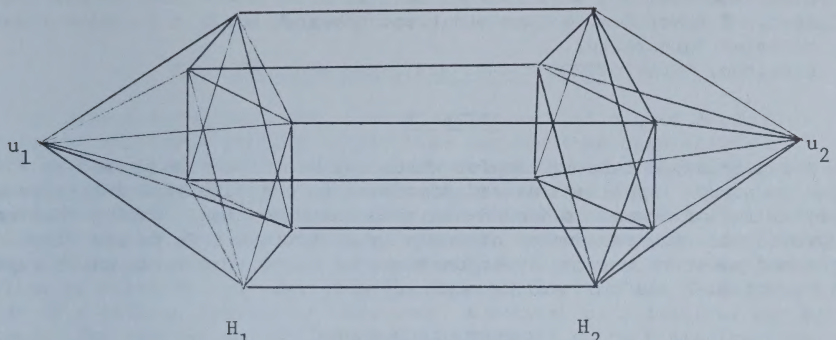


Figure 1: An H(6,3)-extremal graph.

CODING THEOREMS FOR SUPERPOSABLE CODES

J.B. ANDERSON
Dept. of Elec. Eng. & Communications Research Lab.
McMaster University,
Hamilton, Ontario.

ABSTRACT

A superposable code is one for which any N or fewer words add to a unique sequence. Applications are discussed to code-division multiplex and to signalling of demand in demand-assigned communication. Coding theorems are proved relating the number of words in an optimum code to its block length and power to resolve superpositions of words, for words which superpose by modulo-2 addition and by ordinary addition.

I. SUPERPOSABLE CODING

Consider a code, for which any N or fewer words add to a unique sequence. Up to N transmitters may send coded information through the same additive medium, and a receiver, using the uniqueness of sums, can still decode the superposition into separate messages.

Such superposable codes have clear application to multiple access communication. To visualize the type of problem they are designed to solve, consider the two networks of Fig. 1. Typical of satellite multiple-access systems, the star network has stations arranged around a central node, which is either the satellite or reached solely through it. The second topology, dual to the first, consists of various stations as well as the central control, spaced around a single loop. Digital loops are finding application in telephone and data communication.

Now suppose that stations A and B in either topology simultaneously transmit binary sequences to the central control, for instance, the sequences 1100 (A) and 0110 (B). The sequences will add by a mechanism characteristic of the multiple access channel. At worse, they will destructively interfere, but with proper design, they may interfere in a meaningful way. In a satellite star network, on/off signalling without limiters (among other methods) will lead to ordinary addition, and in the example the central node receives the superposition 1210. To decode this sequence, the node must consider the possible pairs of sequences which add to 1210. Many pairs are possible, and even if the correct pair are decided, it remains ambiguous which sequence came from which station.

In the loop topology, the natural addition rule is modulo-2 addition. In a loop with bipolar modulation, for example, each station can observe the bit stream as it flows by, complementing those bits where its own superposed bit is a 1. Now the superposition in the example is 1010, which, as before, will be subject to many possible decompositions. Two special examples are 1000 + 0010, the pair corresponding to time-division multiplex, and 0000 + 1010, which results when A sends nothing at all.

From its definition, it is clear that a superposable code with N=2 is what is required to decompose uniquely any superposition received at the central control, providing all transmitted sequences are code words. The

---

* This work supported in part by National Research Council of Canada, Grant A8828.

ambiguity of who sent what may be resolved by assigning certain parts of the code to each user. Theorems can be written outlining the asymptotic properties of such codes, with various implications for multiple access communication, but first we investigate two systems which use the codes.

II. TWO USES FOR SUPERPOSABLE CODES

In demand-assigned communication, stations must signal demand for service to a central control, which then assigns them channel space. Stations have no means of communication except through the multiple access channel. Unless they signal in exclusively reserved slots ("polling"), demands for service will occasionally interfere when several stations unwittingly signal at once. For many systems, interference is sufficiently rare, but many users or short messages will both reverse this and make polling prohibitive. One solution is superposable coding. Using the S words of a code as identifier sequences, a network of S stations may be served. The central control correctly interprets up to N simultaneous demands for service. It has been shown that such a system can hugely improve on polling the stations, and for channels which add mod-2, the scheme is an asymptotically optimum [2] method of communicating demand.

As a second example, suppose N transmitters simultaneously use the same channel. Subdividing a superposable code into N subcodes of S/N words, let each transmitter use one. Each will transmit at rate log (S/N), and a receiver will exist to decode correctly any one or all of the N separate messages. It is worth noting that the receiver decodes information of two types, what the messages are, and which transmitter they come from. Demand signalling is the extreme case of S stations and 1-bit messages.

III. SUMMARY OF CODING THEOREMS

The multiple access channel determines the mechanism of interference, and with an addition rule established, one can ask how large a code exists for a fixed block length ℓ, that can resolve all superpositions of N or fewer words. We show for ordinary addition that the number of words S satisfies asymptotically

$$\ell \approx \frac{2N \log S}{\log (\pi N)} \tag{1}$$

For mod-2 addition, this separating power is much less, only

$$\ell \approx N \log S \tag{2}$$

In either case, assume throughout that code word letters are chosen from the same alphabet $\{0,1\}$, (although channel symbols in the first case will be from $\{0,\ldots,N\}$). (1) and (2) apply for fixed $N \ll S$.

One can also show relations for $N \approx \lambda S$, $\lambda < 1$. In the demand-signalling example, requests for service are proportional to the number of stations, and λ is the signalling probability for each of S identical stations. More generally, consider any system in which the number of superposing signals is proportional to the number of stations. For ordinary addition, (1) becomes for small λ

$$\ell \approx \frac{2S\, h(\lambda)}{\log (\pi \lambda S)} \tag{3}$$

(h( ) is the binary entropy function), and for modulo-two addition (2) is

$$\ell \approx S\, h(\lambda) \tag{4}$$

It is clear from (1) - (4) that channels with different addition mechanisms vary widely in their ability to carry superposed information. For the same number of words, ordinary addition requires a factor 1/2 log(πN) shorter block length than mod-2 addition. However, a second channel cost factor will be the cost of modulation, and a weaker superposition rule may be realizable with far cheaper modulation. In reality, system design will be a complicated trade-off between the cost of modulation and the coding power it will support.

IV. CODING THEOREMS FOR MOD-2 SUPERPOSITION

To begin, we must distinguish between codes which resolve unambiguously every superposition of N or fewer words, and those which resolve "enough", but not quite all, superpositions. The first attitude is similar to that of error correcting codes, where all combinations of t or fewer errors are corrected. The second parallels the standard channel coding theorem. For each $n \leq N$, the code distinguishes all superpositions of n words from other superpositions of N or fewer words, save a fraction which tends to zero as the code size grows. If the various superpositions are reasonably distributed in probability, for instance, if all n-fold superpositions are equally likely, then the probability of accurate information transmission tends to one as the code size grows. As with ordinary channel coding, this latter attitude leads to much increased transmission.

Theorem 1 establishes the power of mod-2 superposition for the case $N \approx \lambda S$, under the second attitude. The proof, for which only a sketch is necessary, also demonstrates the link between superposable codes of resolving ability λS and ordinary linear codes used with a binary symmetric channel of error probability λ. The theorem has appeared in [1].

Theorem 1

Let $N \approx \lambda S$. In the limit of large S, a code exists that correctly distinguishes superpositions of $\lesssim \lambda S$ words with probability $\varepsilon \downarrow 0$, assuming the words are reasonably distributed, where

$$\ell \approx S\, h(\lambda) \tag{5}$$

Here and throughout, the notation $\approx$ means asymptotically equal in ratio.

Direct Proof

Consider a vector of length S whose i-th digit is 1 if the i-th code word is part of the superposition, and 0 otherwise. Now consider the H-matrix of a linear code of length S. From algebraic coding theory, the product of this vector and H, called the syndrome, allows determination of the 1's in the vector, called the error pattern; further, there exists an H of dimension k x S enabling recognition of patterns of $\lesssim \lambda S$ 1's with probability $\varepsilon \downarrow 0$ as $S \to \infty$, provided that $k \approx S\, h(\lambda)$. If the i-th code word is simply the i-th column of H and mod-2 addition of words occurs, the superposed sequence is the syndrome, from which the receiver can decode the individual words.

Converse Proof

Asymptotically, the number of combinations of S words taken λS or fewer at a time is $\approx 2^{Sh(\lambda)}$. To represent each of these with a different bit sequence requires at least $\approx Sh(\lambda)$ bits.

QED

We see that linear code that corrects N errors provides through the columns of its H matrix the words of a superposable code that resolves N-fold superpositions (and vice versa). Statements about error correcting codes of block length S, rate $(S-\ell)/S$ and distance 2N+1 correspond directly to statements about superposable codes with parameters S, ℓ, and N. A perfect code, for instance, generates a "perfect" superposable code with S words of length ℓ and separating ability N, having the least block length possible for its S and N. A similar statement holds for the quasi-perfect codes; in particular the 2-error correcting BCH codes (which are quasi-perfect) provide an optimal class of 2-fold resolving superposable codes. For general S, ℓ, and N, a code catalog such as that of Sloane [3] provides the best known combinations. Fig. 2 plots the maximal log S taken from [3] for each block length, at N = 2-5.

As shown in [2], one can also bound the power of mod-2 coding when every superposition of N or fewer words is unique, in a result parallel to the Gilbert bound.

Turning to fixed N and arbitrary S, it is straightforward to show using the method of the Appendix that

$$\ell \approx N \log S \tag{2}$$

But (2) stems more basically from the code word and channel symbol alphabets being identical. Consider the second example of II. If the N transmitters each send log S bits/frame and merely time-division multiplex a frame of ℓ symbols, S, ℓ, and N will relate as in (2). Since each transmitter's set of log S bits can be i.i.d., no coding scheme can improve on (2).

V. CODING THEOREMS FOR ORDINARY ADDITION

Consider code words consisting of 1's and 0's, as in the previous section, but let the superposition occur by ordinary addition. Theorem 2 gives the coding power of this superposition rule.

Theorem 2

Let N be fixed and the number of code words S grow. There exists a code of length

$$\ell \lesssim \frac{N \log S - \log N!}{\log(1/\rho)} \tag{6}$$

which distinguishes all superpositions of n words from all other superpositions of $\leq$ N words, except for a fraction N/log S, n = 1,...,N. Here

$$\rho(N) = 2^{-2N} \sum_{k=0}^{N} \binom{N}{k}^2 \tag{7}$$

Proof: See Appendix

$\rho(N)$ is a fundamental quantity, the probability that the i-th letter of one N-fold superposition is equal the i-th letter of some other N-fold superposition, none of whose component code words are the same. Since the proof of Theorem 2 proceeds by random coding, code word letters are chosen i.i.d., and (7) follows directly. Except for small N, an easy proof (see Feller [4], pg. 75) shows that $\rho(N)$ tends to zero like $1/\sqrt{\pi N}$. Setting aside other lower order terms, one gets for (6)

$$\ell \lesssim \frac{2N}{\log(\pi N)} \log S \tag{1}$$

To prove Theorem 2, we use a method of indicator functions, combined with the union bound of probability, and choose code words at random. The same method proves Theorem 1 (note $\log(1/\rho) = 1$ for all N), but would not have offered the analogy to error correcting codes. In the proof, it develops that the best bound (5) occurs when code word letters are chosen 0 or 1 with probability 1/2. This suggests that the best binary superposable codes have words whose digits approximately balance. Intuition, on the other hand, seems to say that lighter weight codewords have less difficulty superposing with each other. The words inherent in time-division multiplex have only a single 1 per word. Nonetheless, our searches for actual codes (see [2]) bear out the theorem conclusion. Whether or not letters should be equiprobable in the non-binary case is an open question. It is possible that code words using an (N+1)-level channel, as ours do, could utilize (N+1)-level letters to achieve a more powerful performance than (1).

When $N \approx \lambda S$, a fixed fraction of S, the indicator method yields (3).

Theorem 2A.

Let $N \approx \lambda S$. For small λ, there exists a code in the limit of S having

$$\ell \lesssim \frac{2S\ h(\lambda)}{\log[\pi(\lambda + O(\lambda^2))S]} \tag{3}$$

Proof: See [2]

Comparing bounds (1)-(4), we see that a code of S words requires about $1/2 \log(\pi N)$ times more symbols for either fixed N or $N \approx \lambda S$, if superposition occurs by mod-2 instead of ordinary addition. This stems clearly from the higher capacity of the (N+1)-level noiseless channel.

We have assumed that (1) and (3) are not bounds, but actual estimates. Further study of the ordinary-addition channel shows a converse to (1) and (3) in the following sense. If N transmitters choose words with i.i.d. letters, the letters sum to form binomially distributed channel symbols, whose entropy, as an easy lemma shows, is about $1/2 \log(\pi N)$ bits/symbol for large N. But (1) implies that at least this many bits must be invested in each symbol. One can view decoding a superposition as determining which sum has been transmitted out of the possible sums of S words. Asymptotically, there are $\approx S^N/N!$ such sums, which implies that $\approx$ N log S bits of information have passed through the channel in ℓ symbols, or by (1), about $1/2 \log(\pi N)$ bits/symbol. A similar argument applies to (3); here there are $\approx 2^{Sh(\lambda)}$ sums to distinguish.

Assuming that no superposable code can effect the transfer of more than $1/2 \log(\pi N)$ bits, we see that in the limit of N, superposable codes use about 1/2 the capacity of the noiseless (N+1)-level channel (whose capacity is log(N+1) bits/symbol). Apparently, this is a penalty inherent in multiple-access communication. The ratio lies closer to unity for small N.

REFERENCES

1. J.B. Anderson, "Optimum Signalling of Demand in a Demand-Assigned Multiple-Access Communication System", Electronics (Lett.), vol. 10, No. 14, pp. 270-271, 11 July 1974.

2. J.B. Anderson, "Theorems for Superposable Codes with Application to Multiple Access Communication", in submission, IEEE Trans. Information Theory, also available as CRL Internal Report Series CRL-33, Communications Research Laboratory, McMaster University, Hamilton, Ontario.

3. N.J.A. Sloane, "A Survey of constructive Coding Theory, and a Table of Binary Codes of Highest Known Rate", Discrete Mathematics, vol. 3, No. 1-3, pp. 265-294, Sept. 1972.

4. W. Feller, An Introduction to Probability Theory and Its Applications, vol. I, 3rd Ed., Wiley, New York, 1968.

APPENDIX

Proof of Theorem 2

Choose S words of length ℓ, $s_1,\ldots,s_S$, choosing each digit i.i.d. with $P_1=p$ and $P_0=1-p$. Let $w_\alpha^{(n)}$ be the α-th sum of n s_i all different. Define the indicator function

$$\phi_\alpha^{(n)} \left\langle \begin{array}{l} 1 \text{ if } w_\alpha^{(n)} \neq \text{ any other superposition of } \leq N \\ \quad s_i \text{ all different} \\ 0 \text{ otherwise} \end{array} \right.$$

where $n = 1,\ldots,N$. If there are W_n n-fold sums of which V_n are unambiguous,

$$E[V_n] = E[\sum_\alpha \phi_\alpha^{(n)}] = W_n \, E[\phi_1^{(n)}]$$

Now calculate $E[\phi_1^{(n)}]$, first for n=N.

$$E[\phi_1^{(N)}] = 1-P_r \left\{ \begin{array}{l} w_1^{(N)} = w_\beta^{(n)} \text{ for at least} \\ \text{one } \beta \text{ and } n \ (\beta \neq 1 \text{ for } n = N) \end{array} \right\}$$

$$\geq 1- \sum_{\beta,n} P_r\{w_1^{(N)} = w_\beta^{(n)}\} ,$$

where we have used the union bound and summed over all such β and n. The terms in this sum are of two types:

i) $\binom{S-N}{M}$ terms for which the M-fold superposition

$w_\beta^{(M)}$ has no s_i in common with $w_1^{(N)}$, $M=1,\ldots,N$.

ii) $\binom{N}{K}\binom{S-N}{M-K}$ terms for which $w_\beta^{(M)}$ has k s_i

in common with $w_1^{(N)}$, $k = 1,\ldots,M$ and $M = 1,\ldots,N$.

Regarding ii), observe that $w_1^{(N)}$ equals another sum having common s_i if and only if the sums are equal with the common s_i removed.

Now use Lemma 1 and definition A6 to write the probability sum as

$$\sum_{M=0}^{N}\binom{S-N}{M}P^{\ell}_{N,M} + \sum_{M=1}^{N}\sum_{k=1}^{M}\binom{N}{k}\binom{S-N}{M-k}P^{\ell}_{N-k,M-k} \qquad \text{(A1)}$$

where we have set $p = \frac{1}{2}$. (Define $P_{0,0} = 0$). By Lemma 1, the first term is overbounded by $P^{\ell}_{N,N}\sum_{M=0}^{N}\binom{S-N}{M}$, while the second is negligible compared to the first by Lemma 2, as $S \to \infty$. Define

$$\rho(N) \triangleq P_{N,N} \qquad \text{(A2)}$$

Then $E[\phi_1^{(N)}] \geq 1 - (C/\log S)^2$ if

$$\frac{C}{\log S}^{2} \gtrsim P^{\ell}_{N,N}\sum_{M=0}^{N}\binom{S-N}{M} \approx \frac{S^N}{N!}\rho^{\ell} \qquad \text{(A3)}$$

(A3) is true as $S \to \infty$ for any C, if

$$\ell \lesssim \frac{N\log S - \log N!}{\log(1/\rho)}(1-\varepsilon) \qquad \text{(A4)}$$

for any $\varepsilon > 0$.

Turning to $\phi_1^{(n)}$, $n < N$, a similar argument holds, with n replacing N (except in the summations over M). Furthermore, the expression $P^{\ell}_{N,N}\sum_{M=0}^{N}\binom{S-N}{N}$ continues to dominate the first terms of (A1) with $n < N$, as an element-by-element comparison shows. Thus (A4) continues to guarantee that $E[\phi_1^{(n)}] \geq 1 - (C/\log S)^2$.

For each n, let $X_n = 1 - V_n/W_n$, the fraction of unuseable n-fold superpositions. By the foregoing, $EX_n < (C/\log S)^2$, and in general,

$$P_r\{X_n \geq A\} \leq E[X_n]/A$$

so that

$$P_r\{X_n \geq \frac{C}{\log S}\} \leq \frac{C}{\log S}$$

Now

$$P_r\{X_n < \frac{C}{\log S}, \text{ all } n\} \geq 1 - \sum_{n=1}^{N} P_r\{X_n \geq \frac{C}{\log S}\},$$

$$\geq 1 - \frac{N}{\log S} \to 1.$$

choosing C = 1. This proves the theorem.

A moment's thought shows the choice of $p = \frac{1}{2}$ to be optimal. An easy lemma shows the dominant first term in (A1) is minimized for $p = \frac{1}{2}$. Choice of another p increases this term and possibly reverses its dominance over the others, either of which leads to a poorer bound (A4).

Lemma 1

Let code word letters be chosen i.i.d. with $P_0 = P_1 = \frac{1}{2}$ (as assumed in the theorem proof).

Define

$$P_{N,M} \triangleq P_r\{\text{Sum of N letters} = \text{Sum of M letters}\} \tag{A6}$$

where the sums are chosen independently. Then

i) $P_{N,M} = 2^{-M-N} \sum_{i=0}^{I} \binom{N}{i}\binom{M}{i}, \quad I = \min(N,M)$

ii) $P_{N,M} = P_{M,N}$

iii) $P_{N,N} \geq P_{N,N+k}\ ; \ k = -N, \ldots -1, 0, +1, 2, \ldots$

Proof

For i) and ii), calculate the probability of sets of N and M Bernoulli trials having an equal number of successes.

iii) follows by manipulation of binomial coefficients.

Lemma 2

Let $S \to \infty$, with N fixed. For $k \geq 1$,

$$\frac{P^{\ell}_{N-k,N-k} \quad S^{N-k}}{P^{\ell}_{N,N} \quad S^{N}} \to 0$$

when $\ell \approx N \log S/\log \sqrt{\pi N}$.

Proof: See [2]

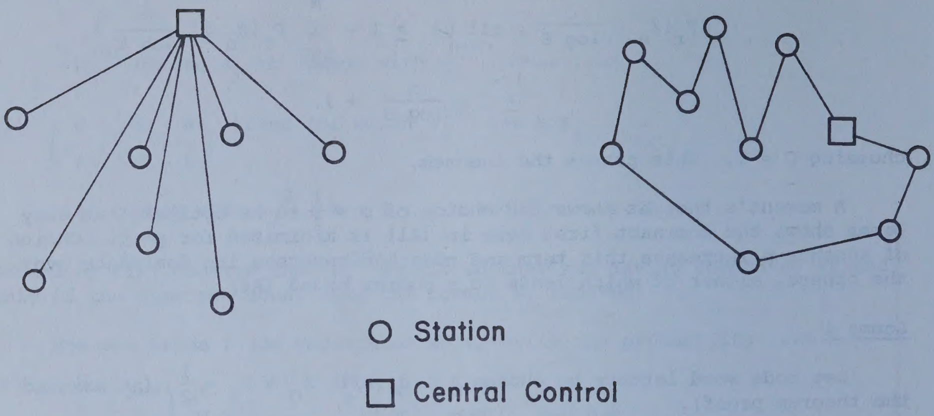

Station
Central Control

Figure 1: Star and Loop Networks

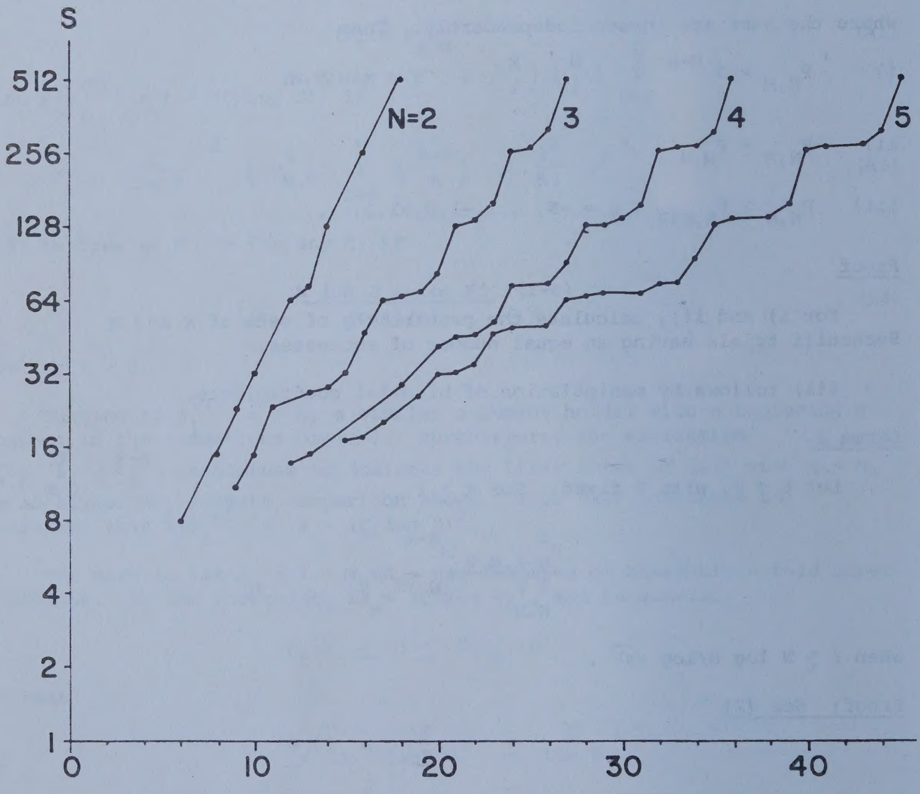


Figure 2: Largest codes known for block length ℓ and resolving ability N, modulo-2 addition; data from [2] (linear codes only).

SUB-BAUD CODING

J. L. MASSEY and J. J. UHRAN
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

ABSTRACT

In a spread-spectrum communication system, a data sequence of +1's and -1's is expanded into a sequence of $+\underline{x}$'s and $-\underline{x}$'s where $\underline{x}$ is a length n sequence of +1's and -1's. A sub-baud code is the assignment of a different such baud sequence $\underline{x}$ to each of M users with the objective of reducing interference between users. It is shown that this interference can be described in terms of the even and odd correlation functions between pairs of sequences. A class of "auto-optimal" sub-baud codes, based on PN sequences, is defined and shown, by example, to be of fair quality. A technique for deriving sub-baud codes from cyclic error-correcting codes is given, and a theorem proved which bounds the quality of the resultant code. A specific such cyclic code construction procedure is given which, for any M, yields a sub-baud code for which the mutual interference can be made arbitrarily small by choosing n sufficiently large. This is believed to be the first such constructive procedure, although recent work by Schneider and Orr proved the existence of such "arbitrarily good" sub-baud codes.

I. INTRODUCTION

Consider a binary spread-spectrum communication system of the following type. Let ... $b_{-1}, b_0, b_1, \ldots$, where b_i is +1 or -1, denote the data sequence to be transmitted. Let $\underline{x} = [x_0, x_1, \ldots, x_n]$, where x_i is +1 or -1, be the fixed sequence used for bandwidth expansion in the manner that $\ldots, b_{-1}\underline{x}, b_0\underline{x}, b_1\underline{x}, \ldots$ forms the actual transmitted sequence of digits. In other words, the digit sequence in the baud controlled by b_i is $+\underline{x}$ or $-\underline{x}$ according as b_i is +1 or -1. A correlation receiver recovers the data sequence by correlating with $\underline{x}$ over each baud since, in the absence of noise,

$$(b_i\underline{x})\cdot\underline{x} = nb_i. \qquad (1)$$

However, when the receiver is k digits $(0 \leq k < n)$ out-of-phase with the baud edges, its output is given by

$$(b_{i-1}[x_{n-k}, \ldots, x_{n-1}], b_i[x_0, \ldots, x_{n-k-1}])\cdot\underline{x} = b_{i-1}F_{n-k} + b_iF_k \qquad (2)$$

where

$$F_k = \sum_{h=0}^{n-k-1} x_h x_{h+k}, \qquad 0 \leq k < n, \qquad (3)$$

is the usual non-periodic correlation function [1] of the sequence $\underline{x}$.

For $k \neq 0$, it is desirable that the magnitude of the righthand side of (2) be small compared with n to permit reliable baud synchronization and also, in certain applications, to reject multipath delayed versions of the transmitted signal. Thus, we are motivated to consider the two correlation functions

$$E_k = F_k + F_{n-k} \qquad (4)$$

and

$$\theta_k = F_k - F_{n-k} \qquad (5)$$

whose magnitudes are the possible values of the righthand side of (2). E_k is the usual "periodic correlation function" [1] for $\underline{x}$, but θ_k appears not to have been previously studied. We note from (4) and (5) that

$$E_{n-k} = E_k \tag{6}$$

but that

$$\theta_{n-k} = -\theta_k \tag{7}$$

and, hence, we shall call E_k and θ_k the even and odd crosscorrelation functions of $\underline{x}$, respectively.

With the sequence $\underline{x}$, we associate a binary vector $\underline{v} = [v_0, v_1, \ldots, v_{n-1}]$ in the manner that v_i is 0 or 1 according as x_i is +1 or -1. Let T denote the usual cyclic shift operator, i.e., $T\underline{v} = [v_{n-1}, v_0, v_1, \ldots, v_{n-2}]$. Let N denote the "compcyclic shift" operator which we define by $N\underline{v} = [\bar{v}_{n-1}, v_0, v_1, \ldots, v_{n-2}]$ where the overbar denotes binary complementation. Notice that, for $0 \le k < n$,

$$N^k\underline{v} = T^k\underline{v} + \underline{B}_k \tag{8}$$

where $\underline{B}_k = [1,1,\ldots,1,0,0,\ldots 0]$ is the "solid burst" with k leading 1's, and where here and hereafter all operations on binary vectors are carried out in modulo-two arithmetic.

If $\underline{x}$ and $\underline{x}'$ are two ± 1 sequences having $\underline{v}$ and $\underline{v}'$ as their corresponding binary vectors, we can write

$$\underline{x} \cdot \underline{x}' = n - 2d(\underline{v}, \underline{v}'), \tag{9}$$

where d is the Hamming distance function, as follows from the fact that each position i where $x_i \neq x_i'$ subtracts 2 from the maximum correlation n achieved when $\underline{x} = \underline{x}'$. Making use of (9) in (2) with $b_{i-1} = b_i = +1$, we see from (4) that

$$E_k = n - 2d(\underline{v}, T^k\underline{v}). \tag{10}$$

Similarly, taking $b_{i-1} = -1$ and $b_i = +1$, we see from (5) that

$$\theta_k = n - 2d(\underline{v}, N^k\underline{v}). \tag{11}$$

It is well-known that the even correlation function is invariant to cyclic shifting. This can be proved easily from (10), for if $\underline{v}' = T\underline{v}$ then $E_k' = n - 2d(\underline{v}', T^k\underline{v}') = n - 2d(T\underline{v}, T^{k+1}\underline{v}) = n - 2d(\underline{v}, T^k\underline{v}) = E_k$. However, the odd correlation function depends strongly on the "phase" of the sequence as the following example shows:

Example:

Let $\underline{v}$ be the n = 7 PN sequence [1,0,0,1,1,1,0] (a PN sequence is one period of the output of a maximal-length linear feedback shift-register.) As for all PN sequences, $E_k = -1$ for $k \neq 0$. The odd correlation function is

k	1	2	3	4	5	6
θ_k	+1	-5	-3	+3	+5	-1

which shows the poor quality of this sequence. However, the PN sequence $\underline{v}' = T\underline{v} = [0,1,0,0,1,1,1]$ has

k	1	2	3	4	5	6
θ'_k	+1	-1	+1	-1	+1	-1

which is the best possible autocorrelation function. [This shift of the original PN sequence happens to be a Barker sequence [1], i.e., $|F_k|$ is 0 or 1 for all $k \neq 0$, which accounts for the excellence of both its even and odd autocorrelation functions as can be seen from (4) and (5).]

Consider now a spread-spectrum communication system with M users. By a sub-baud code for such a system we mean the M binary n-tuples $\underline{v}^{(1)}$, $\underline{v}^{(2)}$, ..., $\underline{v}^{(M)}$ which correspond to the users' baud sequences $\underline{x}^{(1)}$, $\underline{x}^{(2)}$, ..., $\underline{x}^{(M)}$. We now define the even and odd crosscorrelation functions between pairs of codewords as

$$E_k^{(i,j)} = F_k^{(i,j)} + F_{n-k}^{(i,j)} \tag{12}$$

and

$$\theta_k^{(i,j)} = F_k^{(i,j)} - F_{n-k}^{(i,j)}, \tag{13}$$

respectively, where

$$F_k^{(i,j)} = \sum_{h=0}^{n-k-1} x_h^{(i)} x_{h+k}^{(j)} \tag{14}$$

is the usual non-periodic crosscorrelation function between $\underline{x}^{(i)}$ and $\underline{x}^{(j)}$.

An analysis, similar to that given above for the autocorrelation case, shows that the magnitudes of $E_k^{(i,j)}$ and $\theta_k^{(i,j)}$ are the possible values of the output from the i-th receiver's correlator when the received signal is that from the j-th user and when the correlator is k digits out-of-phase with the baud edges. Of course, $E_k^{(i,i)}$ and $\theta_k^{(i,i)}$ are just the autocorrelation functions for $\underline{x}^{(i)}$ which we denote hereafter as $E_k^{(i)}$ and $\theta_k^{(i)}$. The even crosscorrelation function is well-studied [2,3] and is often called the periodic crosscorrelation function. However, in spite of its equal importance in the spread-spectrum communication problem, the odd crosscorrelation function appears not to have been previously studied. It can be easily shown that

$$E_k^{(i,j)} = n - 2d(\underline{v}^{(i)}, T^k\underline{v}^{(j)}) \tag{15}$$

and

$$\theta_k^{(i,j)} = n - 2d(\underline{v}^{(i)}, N^k\underline{v}^{(j)}) \tag{16}$$

from which the following symmetry relations can be easily obtained:

$$E_{n-k}^{(i,j)} = E_k^{(j,i)} \tag{17}$$

$$\theta_{n-k}^{(i,j)} = -\theta_k^{(j,i)}. \tag{18}$$

The objective in sub-baud coding is to minimize the spurious contributions to the outputs of the correlation receivers for the M users, i.e., to minimize the maximum magnitude of all the "off-peak" (i.e., $k \neq 0$) values of all the autocorrelation functions and to minimize the maximum magnitude of all the values of all the crosscorrelation functions More specifically, the objective of sub-baud coding is to make small, compared with n, each of the following maximum "spurious" correlations:

$$\mu_{Ea} = \max_{1\le i\le M} \max_{1\le k<n} |E_k^{(i)}| \tag{19}$$

$$\mu_{\theta A} = \max_{1\le i\le M} \max_{1\le k<n} |\theta_k^{(i)}| \tag{20}$$

$$\mu_{EC} = \max_{1\le i<j\le M} \max_{0\le k<n} |E_k^{(i,j)}| \tag{21}$$

$$\mu_{\theta C} = \max_{1\le i<j\le M} \max_{0\le k<n} |\theta_k^{(i,j)}| \tag{22}$$

where the subscripts E, θ, A and C denote "even", "odd", "autocorrelation", and "crosscorrelation", respectively.

II. AUTO-OPTIMAL PN SUB-BAUD CODES

Because every PN sequence has $E_k \equiv -1$ for $1 \le k < n$, a sub-baud code composed entirely of PN sequences has $\mu_{EA} = 1$ so that such "PN sub-baud codes" appear attractive at first blush. A PN-sequence is one period of a non-zero binary sequence generated by a maximal-length linear feedback shift-register (LFSR), i.e., a LFSR whose connection polynomial is a primitive polynomial of degree L over GF(2) where L is the register length [4]. The period of a non-zero output sequence of a maximal-length LFSR is $n = 2^L-1$ so that the 2^L-1 PN sequences generated by a given maximal-length LFSR are all cyclic shifts of the same sequence. There are $\phi(2^L-1)/L$ different primitive polynomials of degree L, where ϕ is Euler's "phi function", so there are just this many cyclically distinct PN sequences of length $n = 2^L-1$. For example, there are $\phi(2^7-1)/7 = 18$ cyclically distinct PN sequences of length $n = 2^7-1 = 127$. We define a PN sub-baud code to be a sub-baud code with $n = 2^L-1$ and $M = \phi(2^L-1)/L$ whose codewords are cyclically distinct PN sequences of length n.

Because the odd correlation properties of a sub-baud code depend strongly on the "phase" of each sequence, there are a multitude of non-equivalent PN sub-baud codes for any given n, viz. the n^M codes corresponding to the n possible choices of "phase" for each of the M cyclically distinct PN sequences. For example, there are $(127)^8 \approx 7 \times 10^{16}$ non-equivalent PN sub-baud codes of length n = 127. A natural way to reduce this bewildering amount of freedom is by choosing (one of) the best phase(s) of each sequence in the sense of minimizing the maximum off-peak magnitude of its odd autocorrelation function and (when several phases yield the same maximum) the number of occurences of this maximum. We call a PN sub-baud code auto-optimal when each of its component sequences has optimum "phase" in this sense.

To ascertain the quality of auto-optimal PN sub-baud codes, we made a computer search for such codes for the case n = 127. Somewhat surprisingly, we found there was a unique auto-optimal code at this length, i.e., there was a unique "best" phase for each of the 18 cyclically distinct sequences.

For 2 of these "best" PN sequences, the maximum off-peak magnitude of θ_k was 15, for 12 the maximum was 17 and for 4 the maximum was 19. Thus,

$$\mu_{\theta A}/n = 19/127 = .150 \tag{23}$$

for this sub-baud code. A check of the crosscorrelation properties of this code revealed that

$$\mu_{EC}/n = \mu_{\theta C}/n = 41/127 = .323. \tag{24}$$

For completeness, we state explicitly that

$$\mu_{EA}/n = 1/127 = .008. \tag{25}$$

From this example, we see that the excellence of the even autocorrelation functions of a PN sub-baud code, as in (25), is paid for by the only fair crosscorrelation functions of both even and odd type, as in (24), and the only moderately good odd autocorrelation functions, as in (23). The fact that $\mu_{EC} = \mu_{\theta C}$ for this particular sub-baud code does suggest, however, that good odd correlation properties may be intrinsically no more difficult to achieve than good even correlation properties.

III. SUB-BAUD CODES DERIVED FROM CYCLIC CODES

We now turn to a more systematic construction of sub-baud codes, employing the tools of cyclic binary error-correcting codes as given, for instance, in [5] or [6].

The cyclic order of the n-place binary vector $\underline{v}$ is the smallest positive integer ν such that $T^{\nu}\underline{v} = \underline{v}$. The cyclic equivalence class (CEC) containing $\underline{v}$ is the set of vectors which are all ν distinct cyclic shifts of $\underline{v}$. "Cyclically distinct" vectors, as described in the previous section, are just vectors in different CEC's.

With the vector $\underline{v} = [v_0, v_1, \ldots, v_{n-1}]$, one can associate the polynomial $v(x) = v_0 + v_1x + \ldots + v_{n-1}x^{n-1}$ of degree less than n. The cyclic code generated by g(x), a divisor of $x^n + 1$, is the set of $\underline{v}$ such that g(x) divides v(x). The polynomial $h(x) = (x^n + 1)/g(x)$ is called the parity-check polynomial of the cyclic code, and $x^L h(x^{-1})$, where L = n-degree [g(x)], is the connection polynomial of an L-stage LFSR which can serve as an encoder for the cyclic code, i.e., the codewords are the 2^L distinct sequences of length n generated by this LFSR corresponding to the 2^L distinct initial loadings of the register.

Let $W(\underline{v})$ denote the Hamming weight of $\underline{v}$. The minimum weight, W_{min}, of a cyclic code is the minimum $W(\underline{v})$ over all codewords $\underline{v} \neq \underline{0}$. Numerically, W_{min} equals the minimum distance, d, of the code which is defined as the minimum Hamming distance between distinct codewords. We let W_{max} be the maximum weight of the code vectors exclusive of $\underline{1} = [1,1,\ldots,1]$, and note that when $\underline{1}$ is a codeword then $w_{max} = n-W_{min} = n-d$.

With these preliminaries, we now prove:

Theorem 1:

Let C be the length n cyclic code with $h(x) = (x + 1)h_o(x)$, where x+1 is not a factor of $h_o(x)$, and having minimum distance d. Let C_o be the length n cyclic code with parity check polynomial $h_o(x)$. Then, any sub-baud code whose M codewords are selected each from a different CEC of order n among the codewords of C_o satisfies

$$\mu_{EA} \leq n - 2d \tag{26a}$$

$$\mu_{\theta A} \leq n - d + 2 \tag{26b}$$

$$\mu_{EC} \leq n - 2d \tag{26c}$$

$$\mu_{\theta C} \leq n - d + 2. \tag{26d}$$

Proof:

We note first that a cyclic code has $\underline{1}$ as a codeword if and only if x+1 is a factor of its parity check polynomial [because this is the condition for the corresponding polynomial $1 + x + \ldots + x^{n-1} = (x^n + 1)/(x+1)$ to be divisible by $g(x) = (x^n + 1)/h(x)$.] Thus, $\underline{1}$ is a codeword in C but not in C_o. But C_o is a subcode of C and every $\underline{v} \neq 0$ in C_o satisfies

$$d \leq W(\underline{v}) \leq n - d \tag{27}$$

because $W_{min} = d$ and $W_{max} = n - d$ in the larger code C and $\underline{v} \neq \underline{1}$.

Let $\underline{v}^{(i)}$ and $\underline{v}^{(j)}$ be distinct codewords in the hypothesized sub-baud code, i.e., codewords from distinct CEC's of order n in C_o. Then, because

$$d(\underline{v}^{(i)}, T^k\underline{v}^{(j)}) = W(\underline{v}^{(i)} + T^k\underline{v}^{(j)}) \tag{28}$$

and because $T^k\underline{v}^{(j)} \neq \underline{v}^{(i)}$ by the assumed distinctness of the CEC's, it follows that $\underline{v}^{(i)} + T^k\underline{v}^{(j)}$ is a non-zero codeword in C_o and (27) gives

$$d \leq d(\underline{v}^{(i)}, T^k\underline{v}^{(j)}) \leq n - d. \tag{29}$$

Using (29) in (15), we obtain

$$n - 2(n - d) \leq E_k^{(i,j)} \leq n - 2d$$

or, equivalently,

$$|E_k^{(i,j)}| \leq n - 2d. \tag{30}$$

From (30) and (21), (26c) now follows.

Next, we consider

$$d(\underline{v}^{(i)}, N^k\underline{v}^{(j)}) = W(\underline{v}^{(i)} + N^k\underline{v}^{(j)})$$

$$= W(\underline{v}^{(i)} + T^k\underline{v}^{(j)} + \underline{B}_k) \tag{31}$$

where we have made use of (8). Write $\underline{v} = \underline{v}^{(i)} + T^k\underline{v}^{(j)}$ for convenience. Our interest is in $W(\underline{v} + \underline{B}_k)$. From the facts that

$$W(\underline{v} + \underline{B}_k + T\underline{v} + T\underline{B}_k) \leq 2\,W(\underline{v} + \underline{B}_k)$$

and that

$$W(\underline{B}_k + T\underline{B}_k) = 2,$$

it follows that

$$2\ W(\underline{v} + \underline{B}_k) \geq W(\underline{v} + T\underline{v})\ -2.$$

Moreover, $\underline{v} + T\underline{v}$ is a non-zero codeword in C_o so that

$$2\ W(\underline{v} + \underline{B}_k) \geq d - 2. \qquad (32)$$

Using (32)in (31), we obtain

$$2d(\underline{v}^{(i)},\ N^k\underline{v}^{(j)}) \geq d - 2. \qquad (33)$$

Similarly, noting that $W(\underline{v} + \underline{B}_k) = n - W(\underline{v} + \underline{B}_k + \underline{1}) = n - W(\underline{v} + T^k\underline{B}_{n-k})$ and that $2\ W(\underline{v} + T^k\underline{B}_{n-k}) \geq d - 2$, we deduce that

$$d(\underline{v}^{(i)},\ N^k\underline{v}^{(j)}) \leq n - (d - 2)/2. \qquad (34)$$

Using (33) and (34) in (16), we obtain

$$|\theta_k^{(i,j)}| \leq n - d + 2. \qquad (35)$$

From (35) and (20), (26d) now follows.

A slight modification of the above arguments with $i = j$ but $1 \leq k < n$ establishes the bounds (26a) and (26b). Q.E.D.

With the aid of Theorem 1, one can construct a rich variety of sub-baud codes from known cyclic codes, and bound the various maximum spurious distortions using (26). The only previous systematic constructions of sub-baud codes of which we are aware were those of Gold [2,3]. Gold's codes are special cases of the codes constructible by Theorem 1 when C is chosen as a specified BCH code. Gold gave bounds, identical to (26a) and (26c), for his codes, but did not consider the odd correlation functions which can now be bounded by (26b) and (26d).

IV. "ARBITRARILY GOOD" SUB-BAUD CODES

We now construct a specific class of sub-baud codes according to Theorem 1, but improve on the bounds (26b) and (26d) to show that, for given M, all the μ parameters can be made to satisfy $\mu/n \to 0$ as $n \to \infty$ so that an "arbitrarily good" code can be found for any M by choosing n sufficiently large. The possibility of such a construction can be inferred from a recent paper [7] by Schneider and Orr which proved existence theorems for sets of sequences with specified maximum off-peak magnitudes of the non-periodic crosscorrelation functions.

Our construction is perhaps best-explained via an example. Let α be a primitive element of $GF(2^m)$ where m is divisible by 4. Let $m_i(x)$ denote the minimum polynomial of α^i. Choose $h_o(x)$ in Theorem 1 as $h_o(x) = m_1(x)m_r(x)$ where $r = (8/15)(2^m - 1)$. It can be shown that the code C_o contains $2^4 = 16$ CEC's of order $n = 2^m - 1$ so C_o can be used to construct a sub-baud code for any $M \leq 16$ according to Theorem 1. Moreover, it can be readily checked that α^i is not a root of $h(x) = (x + 1)h_o(x)$ for $r < j < 2^m - 1$. Thus, these powers of α are all roots of $g(x)$ and the BCH bound [5,6] applied to C gives

$$d \geq 2^m - 1 - r = (7/15)n.$$

From (26a) and (26c) we then have the bounds

$$\mu_{EA} \leq (1/15)n \qquad \mu_{EC} \leq (1/15)n \tag{36}$$

whereas (26b) and (26d) give the disappointly-large bounds

$$\mu_{\theta A} \leq (8/15)\, n + 2 \qquad \mu_{\theta C} \leq (8/15)n + 2; \tag{37}$$

but one should not be too hasty in assuming that these latter bounds are tight. In fact, we now show that, for all these μ parameters, $\mu/n \to 0$ as $m \to \infty$ (and hence also $n \to \infty$).

We will make use of the result, proved in [8], that the total Hamming Weight NLp of N distinct binary L-tuples satisfies

$$h(p) \geq (\log_2 N)/L \tag{38}$$

where $h(p) = -p \log_2 p - (1 - p)\log_2(1 - p)$.

The bounds (26) were obtained in the proof of Theorem 1 by bounding $W(\underline{v})$ and $W(\underline{v} + \underline{B}_k)$ when $\underline{v}$ is a non-zero codeword in C_o. We now improve on these bounds for large n. For the example considered here, such a $\underline{v}$ is an output sequence of period n from the $L = m + 4$ stage encoding LFSR. Thus, every span of L consecutive digits along $\underline{v}$ (counting cyclic adjacency) must be distinct, or else the period would be smaller than n. Hence, the total weight of these $N = n = 2^m - 1$ L-tuples, NLp, satisfies $h(p) \geq \log_2(N/L) = \log_2(2^m - 1)/(m + 4) \to 1$ as $m \to \infty$ which implies $p \to 1/2$. But this total weight is just $LW(\underline{v})$ because each digit in $\underline{v}$ appears in L such spans, thus

$$W(\underline{v})/n \to 1/2 \quad \text{as } n \to \infty. \tag{39}$$

Using (39) in the proof of Theorem 1 leads to

$$\mu_{EA}/n \to 0 \quad \text{and} \quad \mu_{EC}/n \to 0 \tag{40}$$

as $n \to \infty$. $W(\underline{v} + \underline{B}_k)$ is only slightly more complicated to handle. No L digit span (not counting cyclic adjacency now) within the first k digits or within the last $n - k$ digits of $\underline{v} + \underline{B}_k$ can coincide. The shift-register argument applied to each of these two runs encounters $k \simeq n/2$ as the worst case; but there are then $N \simeq n/2 \simeq 2^{m-1}$ different $L = m + 4$ digit spans in each section. The distinctness argument again leads to

$$W(\underline{v} + \underline{B}_k)/n \to 1/2 \text{ as } n \to \infty. \tag{41}$$

Incorporating (41) into the proof of Theorem 1 leads to

$$\mu_{\theta A}/n \to 0 \quad \text{and} \quad \mu_{\theta C}/n \to 0 \tag{42}$$

as $n \to \infty$.

The reader should now be able to supply the necessary details for the proof of:

Theorem 2:

The sub-baud codes with $M = 2^s$ codewords and length $n = 2^m - 1$, constructed as in Theorem 1 when m is a multiple of s and when $h_o(x) = m_1(x)\, m_r(x)$ where $r = [2^{s-1}/(2^s - 1)](2^m - 1)$, satisfy (40) and (42) as $n \to \infty$.

ACKNOWLEDGMENT

The research reported here was supported by the National Aeronautics and Space Administration under NASA Contract NAS5-10786 and NASA Grant NSG 5025 at the University of Notre Dame in liaison with the Goddard Space Flight Center.

REFERENCES

[1] R. Turyn, "Sequences with small correlation," in Error Correcting Codes (Ed. H. Mann). New York: Wiley, 1968.

[2] R. Gold, "Optimal binary sequences for spread spectrum multiplexing," IEEE Trans. Info. Th., Vol. IT-13, pp. 619-621, Oct. 1967.

[3] R. Gold, "Maximal recursive sequences with 3-valued recursive cross-correlation functions," IEEE Trans. Info. Th., Vol. IT-14, pp. 154-156, Jan. 1968.

[4] J. L. Massey, "Shift-Register Synthesis and BCH Decoding," IEEE Trans. Info. Th., Vol. IT-15, pp. 122-127, Jan. 1969.

[5] W. W. Peterson and E. J. Weldon, Jr., Error-Correcting Codes (2nd edition). Cambridge, Mass: M.I.T. Press, 1972.

[6] E. R. Berlekamp. Algebraic Coding Theory. New York: McGraw-Hill, 1968.

[7] K. S. Schneider and R. S. Orr, "Aperiodic correlation constraints on large binary sequence sets," IEEE Trans. Info. Th., Vol. IT-21, pp. 79-84, Jan. 1975.

[8] J. L. Massey, "On the fractional weight of distinct binary n-tuples," IEEE Trans. Info. Th., Vol. IT-20, p. 131, Jan. 1974.

CODING FOR MULTIPLE-ACCESS COMMUNICATIONS SATELLITES

W. W. WU
COMSAT Laboratories
Clarksburg, Maryland 20734

ABSTRACT

This paper reports the usefulness of channel coding in multiple-access communications satellite applications. Examples of codes used or proposed for use in several satellite systems are presented: the $(n = 63, k = 36, d_{min} = 11)$ code to be used in the MARISAT System; the two block and two convolutional codes used for error detection and error correction, respectively, in the PCM|...|PSK|FDMA systems; and the BCH (31, 21, 5) code in a PCM|PSK|TDMA system. In addition, some proposed coding schemes, including a combined diversity and error control scheme for use in satellite-switched TDMA systems and a selective repeat ARQ scheme for an error detection and retransmission system, are highlighted. Coding for multiple-access interference uses either threshold or Viterbi decoders. Coding for a random multiple-access system is also mentioned.

The intent of this paper is to demonstrate the advantages of coding to communications technologists, and to present some problems to coding theorists.

INTRODUCTION

At present the primary communications satellite mode of operation is the transmission of a preassigned, multidestination, FM carrier with frequency-division multiplexing (FDM) at the baseband and frequency-division multiple access (FDMA) in the RF band. The FDM|FM|FDMA system, which has no coding, uses transponder bandwidth and satellite power inefficiently due to the inflexibility of the channel assignments. Since multiple carriers are transmitted, the satellite transponder cannot operate at full capacity in the FDMA mode due to adjacent channel interference and intermodulation effects. For these reasons other multiple-access techniques and modulation methods have been sought to improve the transmission efficiency of the communications satellite system.

The capacity of a communications satellite system is either power limited or bandwidth limited. Error coding can be used in either case to enhance the quality or reliability of information transmission in a digital satellite system. For small earth stations the capacity is power limited and error coding can be used to increase the capacity by increasing the signal bandwidth. In the band-limited case of INTELSAT earth stations, capacity improvement may also be possible by employing powerful high-rate codes.

Section 2 of this paper gives examples of coding in three multiple-access systems. Section 3 lists some proposed coding schemes for future communications satellite systems, and Section 4 provides some practical coding problems for future research.

This paper is based upon work performed at COMSAT Laboratories under the sponsorship of the Communications Satellite Corporation.

CODING APPLICATIONS IN MULTIPLE-ACCESS SYSTEMS

In this section coding applications in four satellite systems, one military system and three commercial systems, are highlighted. All of these systems are multiple-access systems. In these applications block codes are generally used for control messages, while convolutional codes are used for information messages. The codecs (encoder-decoder pairs) described herein are hardware implementations.

Three codecs have become integral parts of two COMSAT operational systems. Due to system constraints and economic factors, as determined by several tradeoff studies, only two classes of codecs have been used in these systems. For block codes, only BCH codes, derived from either a primitive or a non-primitive element, have been selected for error detection or correction. For convolutional codes, both self-orthogonal codes with short constraint lengths and Viterbi decoding have been used.

Coding in the MARISAT System

MARISAT is a satellite multiple-access system which will provide reliable communications to vessels at sea. The two satellites will be positioned in geostationary orbits, one over the Atlantic and the other over the Pacific Ocean.

The calculated code performance of the $(n = 63, k = 36, d_{min} = 11)$ BCH code for the planned MARISAT System is shown in Figure 1. The bit-error rates for the code used for both error correction and detection are indicated in terms of signal-to-noise ratio (E_b/N_o in dB) in an additive white Gaussian channel.

Laboratory measurements have been performed for several runs. Each run consists of 864×10^6 bits. At channel error rates below 10^{-2}, no undetectable errors have been observed. At higher channel error rates, miss detection (or false detection) synchronization (unique word) errors are dominant. The design error rate of the codec is well within the synchronization error rate of 10^{-5}.

The baseband signaling format of MARISAT is shown in Figure 2. The total frame consists of a unique word (UW) and its inverse ($\overline{UW}$), ship identification (SID), message type and message, TTY for network interfacing, and channel testing for bit-error evaluation (BEE). The message, message type, and SID are coded by the $(n = 63, k = 36, d_{min} = 11)$ BCH code.

The roots of the (63, 36, 11) BCH codes are α^1, α^3, α^5, α^7, and α^9. The minimal polynomials are chosen from degree 6:

$$\begin{aligned} f_1(x) &= 1 + x + x^6 \\ f_2(x) &= 1 + x + x^2 + x^5 + x^6 \\ f_3(x) &= 1 + x^3 + x^6 \\ f_4(x) &= 1 + x^2 + x^3, \text{ and} \\ f_5(x) &= 1 + x + x^2 + x^4 + x^6 \end{aligned}$$

The generator polynomial of the code is

$$\begin{aligned} g(\text{MARISAT}) = 1 &+ x + x^4 + x^8 + x^{15} + x^{17} + x^{18} \\ &+ x^{19} + x^{21} + x^{22} + x^{27} \end{aligned}$$

Figure 3 is the corresponding encoder circuit. Coding in the MARISAT System was suggested by R. McClure and implemented by V. Uzunoglu of COMSAT Labs.

Coding in PCM|PSK|FDMA Systems

There are four operational codecs in this type of low-volume traffic system, which includes the single-channel-per-carrier system. For message transmission there are two convolutional self-orthogonal codes with threshold decoding [1]. The former is a rate 3/4 code of contraint length 80, and the latter is a rate 7/8 code of constraint length 376. The measured results of these codecs are shown in Figure 4 in terms of channel bit-error rate and decoded output bit-error rate.

For message control there is a (73, 55, 5) non-primitive BCH code. The generator polynomial of this code is

$$g(\text{FDMA-Control}) = 1 + x + x^2 + x^3 + x^4 + x^6 + x^9$$

$$+ x^{10} + x^{12} + x^{16} + x^{18}$$

The undetectable bit-error probability is shown in Figure 5.

For data recording there is a (255, 163, 25) BCH code in the terrestrial segment of the PCM|PSK|FDMA system. This powerful codec is used to protect the billing information and to measure the error rate of the transmission link. The code generator polynomial is derived from the primitive polynomial $1 + x^2 + x^3 + x^4 + x^8$. The inverse parity check polynomial, in terms of binary symbol representation, is

$$h(D) = D^{163}\, h(x = 1/D)$$

$$= 1\ 0^3\ 1^3\ 0^2\ 1^7\ 0\ 1\ 0\ 1^2\ 0\ 1\ 0\ 1^2\ 0\ 1^3\ 0\ 1\ 0^2$$

$$1^2\ 0^2\ 1\ 0^4\ 1^3\ 0\ 1^2\ 0\ 1^6\ 0^2\ 1^4\ 0^2\ 1\ 0$$

$$1^2\ 0^2\ 1\ 0^3\ 1^5\ 0^6\ 1^2\ 0^3\ 1\ 0^3\ 1^3\ 0\ 1$$

$$0^4\ 1^2\ 0\ 1^2\ 0^3\ 1\ 0\ 1^2\ 0^3\ 1^2\ 0\ 1\ 0$$

$$1^2\ 0\ 1\ 0\ 1^5\ 0\ 1^2\ 0\ 1\ 0^2\ 1^4\ 0^2\ 1^4\ 0\ 1^2$$

$$0^2\ 1^2\ 0^6\ 1$$

This codec has been implemented in accordance with h(D). It is used for non-real-time detection of errors in digital data which have been transmitted over the link. It is interfaced with the 360/65 computer. The codec has been chosen because it can accommodate the incoming message format, which amounts to 160 bits.

Coding in the PCM|BCH|PSK|TDMA System

The system configuration of the PCM|BCH|PSK|TDMA system is shown in Figure 6. It consists of a 7-bit-per-sample PCM source encoding unit, a BCH (31, 21, 5) channel codec, a preamble generator, and a PSK modem at the signal transmission ends. The baseband signal is processed at 50.0 Mbps. The digital satellite system is a time-division multiple-access system (TDMA).

In Figure 6, F_2 and F_1 are input and output satellite transponder filters, respectively, and F_0 and F_3 represent the up- and down-link filtering processes, which include the output filter of the modulator and the input filter of the demodulator, respectively. The link noises are designated as $n_0(t)$ and $n_1(t)$.

The BCH code is used to enhance the reliability of the bookkeeping information, which consists of the destination code, PCM channel number, and command and control of the system. The generator polynomial of the code is $g(\text{PCM}|\text{BCH}|\text{PSK}|\text{TDMA}) = 1 + x^3 + x^5 + x^6 + x^8 + x^9 + x^{10}$, which is the product of $m_1(x) = 1 + x^2 + x^5$ and $m_3(x) = 1 + x^2 + x^3 + x^4 + x^5$.

The decoder is shown in Figure 7, where a and b are shift registers. Standard symbols are used in the decoder logic circuit. The codec has been field tested successfully with the overall system through the satellite channel.

Coding in a Military TDMA System

A rate 1/2, K = 7 Viterbi decoder has been proposed for an Army TDMA satellite system with a burst rate of 10.0 Mbps [2]. The earth station under consideration has a received power-to-noise ratio of 74.6 dB-Hz with a G/T of 27 dB; the resulting available E_b/N_o is 4.6 dB. For lower speed operation and low rates with less burst switching, the Viterbi decoder is a powerful device for enhancing the quality and reliability in multiple-access communications. The result of simulating the codec performance is shown in Figure 8, which was obtained by S. Lebowitz of COMSAT. This codec is equivalent to Linkabit model LV'7015.

SOME PROPOSED CODING SCHEMES FOR FUTURE MULTIPLE-ACCESS SYSTEMS

This section is intended to briefly describe some proposed coding schemes for possible future communications satellite systems. The first scheme concerns means of improving performance for communications in the presence of fading. The second scheme concerns the improvement of error detection and retransmission system efficiency. The next two schemes concern coding for interference in multiple-access channels. In view of the interference characteristics, coding appears to be very appropriate.

Combined Diversity and Error Control

One way of increasing the capacity of a global communications satellite is by using higher carrier frequencies such as 11/14 or 20/30 GHz in addition to the 4/6-GHz frequencies presently used for transmission. However, at higher frequencies deep fades will be experienced when heavy precipitation is present.

A combined diversity and error control scheme has been suggested by Fang [3]. This scheme uses selection and burst-switching spatial diversity in the up-link and spatial diversity probabilistic and algebraic error control in the down-link of a satellite-switched TDMA multibeam system. Coding is used to improve the satellite link TDMA reliability.

The calculated performance of a dual diversity system is shown in Figure 9. The feasibility of this scheme is presently under evaluation.

Selective Repeat ARQ

One form of coding in communications systems is error detection and retransmission, which is sometimes referred to as an automatic repeat request (ARQ) system. Due to the 600-ms transmission delay through the satellite, conventional start-stop ARQ systems are unsuitable for use over the satellite link.

Gatfield [4] has suggested a selective repeat ARQ system utilizing relatively short code blocks. In terms of transmission efficiency, selective ARQ compares favorably with the stop and wait ARQ system shown in Figure 10. However, in terms of error detection capability, short block codes are not as powerful. Selective repeat ARQ units for satellite applications will be available commercially.

Coding for Interference-Limited Multiple-Access Channel with Threshold Decoding

Measured bit-error rate versus carrier-to-interference (C/I) ratio without coding in a typical satellite system is shown in Figure 11. As C/I increases, the asymptotic leveling off characteristic suggests the advantage of error correcting codes. That is, most codes, when plotted in terms of decoded bit-error rate versus signal-to-noise ratio (SNR) in an additive Gaussian channel, give a larger coding gain at higher values of SNR. This characteristic can be used to advantage to compensate for the leveling off phenomenon due to interference. As an example, the compensated system performance obtained by using the previously described rate 0.875 code [1] for multiple-access interference is also shown in Figure 11. When codes are properly selected for compensating for the effect of interference, the improvement in such system performance can be significant. Such coding has been suggested for a future INTELSAT system by C. Devieux.

Coding for Multiple-Access Interference—Viterbi Decoder

An asynchronous multiple-access communications technique using a combination of orthogonal convolutional coding, on-off signaling, and Viterbi decoding has been described by Cohen, Heller, and Viterbi [5]. The signaling scheme is shown in Figure 12. An information bit interval is partitioned into 2^K equal subintervals, where K is the constraint length of a Viterbi decodable convolutional code. The subintervals are numbered consecutively from 0 to $2^K - 1$. The content of the K-stage shift register after each shift represents a number between 0 and $2^K - 1$. This number is then used to indicate that the corresponding subinterval will contain a pulse, and a 1 is transmitted during this subinterval. The remaining $2^K - 1$ subintervals contain no pulses, i.e., 0's are transmitted,since the 2^K possible signals are time disjointed and thus orthogonal.

A pseudo-random sequence generator is used to randomize the assignment of the subintervals. At each bit time a K-bit portion of the sequence generator is added (mod-2^K) to the contents of the K-stage shift register and the resulting K-bit number is used to generate the pulse signal to be transmitted. At the receiving end a Viterbi encoder has been suggested for decoding such sequences.

The basic approach of the above scheme is similar to that used for the uniform codes described earlier by Massey [6]. The efficiency and capability of the scheme in the presence of non-interference channel noise are not presently known.

Coding for a Random Multiple-Access System

When the number of users is large, e.g., 20,000, as in the earth resources data collection, the effectiveness of both TDMA and FDMA is greatly reduced. This is because the number of frequency divisions in FDMA becomes very high and the required oscillator stabilities impractical. The number of relatively short bursts in TDMA requires overhead for synchronization which may be excessive. In these circumstances, random multiple-access (RMA) systems are appropriate. There are three forms of RMA. That in which PN sequences or frequency hopping techniques with synchronization are used is called code-division multiple access (CDMA). Systems in which retransmission of overlapping messages is permitted are the ALOHA system [8] and its variations [9], [10]. The author [11] has studied a coding algorithm for a third type of RMA system which allows overlapping messages at the satellite without retransmission. Further discussion is beyond the scope of this paper.

RECOMMENDATIONS

The usefulness of coding is severely limited in a noisy channel where error rate improvement sometimes appears to be needed most. For example, at a channel bit-error rate above 5.0×10^{-2}, most codes result in performance which is worse than that without coding. Such high channel error rates are expected in mobile applications in which very small terminals are employed. Codes which can provide reasonable coding gain in a very noisy channel will have an important place in a world of practical coding applications.

Due to the bandwidth limitations of global and domestic commercial communications satellite systems, high-rate codes are essential for error coding to be useful. Practical powerful high-rate codes are expected to have a significant value in the design and operation of future systems of this type.

In addition to the proposed high-speed digital systems, it is anticipated that, for transmission of more information within a given time, high-data-rate transmission is necessary. Thus, the error coding unit must be compatible with the rest of the system, and the decoder must be able to operate at high speeds.

Since multiple-user communications are a primary concern, the complexity and thus the cost of a single codec, which may be multiplied many times, is also an important factor. The cost of an error correcting codec can be traded off against the increasing cost of antenna size, for example.

For severe signal attenuation due to precipitation in higher frequency transmission, error coding has not yet made a significant contribution. The feasibility of developing and evaluating powerful codes to reduce or eliminate the diversity requirements is one of the problems in the design of future satellite systems.

In summary, there is a great deal to be done for coding in multiple-access applications. However, it is felt that the focus of practical coding research should be narrowed to develop powerful high-rate codes, codes for very noisy channels, and high-speed and economical codecs. On the other hand, coding theory should be expanded for the purpose of improving performance by optimizing source coding|channel coding|modulation|multiple-access parameters in a multiple-user communications system.

REFERENCES

[1] W. Wu, "New Convolutional Codes," IEEE Transactions on Communications, Part I, September 1975; Part II, January 1976.

[2] I. M. Jacobs and J. A. Heller, "Performance Study of Viterbi Decoding as Related to Space Communications," Final Technical Report DAAB07-71-C-0418 for the United States Army Satellite Communication Agency.

[3] R. Fang, "DIVEC, A Combined Diversity and Error Control Scheme," COMSAT Technical Review, Vol. 5, No. 1, Spring 1975.

[4] A. G. Gatfield, "ARQ Error Control on the Satellite Channel," Proceedings of the 1974 International Conference on Communications.

[5] A. Cohen, J. Heller, and A. Viterbi, "A New Coding Technique for Asynchronous Multiple Access Communication," IEEE Transactions on Communication Technology, October 1971.

[6] J. Massey, "Uniform Codes," IEEE Transactions on Information Theory, Vol. IT-12, 1966.

[7] "Special Issue on Satellite Communications," Proceedings of IEEE, February 1971.

[8] N. Abramson, "The ALOHA System - Another Alternative for Computer Communications," University of Hawaii Technical Report No. B70-1, 1970.

[9] L. G. Roberts, "Dynamic Allocation of Satellite Capacity through Packet Reservation," National Computer Conference, 1973.

[10] L. Kleinrock and F. A. Tobag, "Carrier Sense Multiple Access for Packet Switched Radio Channels," International Communications Conference, 1974.

[11] W. W. Wu, "A Coding Algorithm for Random Access Satellite Systems," Proceedings of the 1974 International Telemetering Conference.

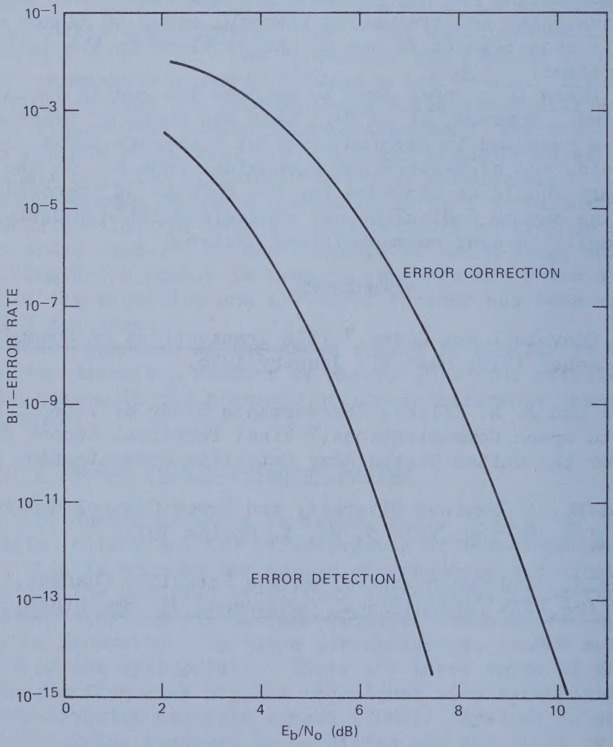


Figure 1. MARISAT Coding Performance

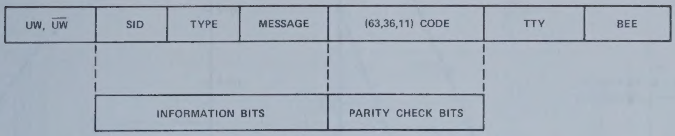

UW, $\overline{UW}$
SID
TYPE
MESSAGE
(63,36,11) CODE
TTY
BEE
INFORMATION BITS
PARITY CHECK BITS

Figure 2. MARISAT Coded Signal Format

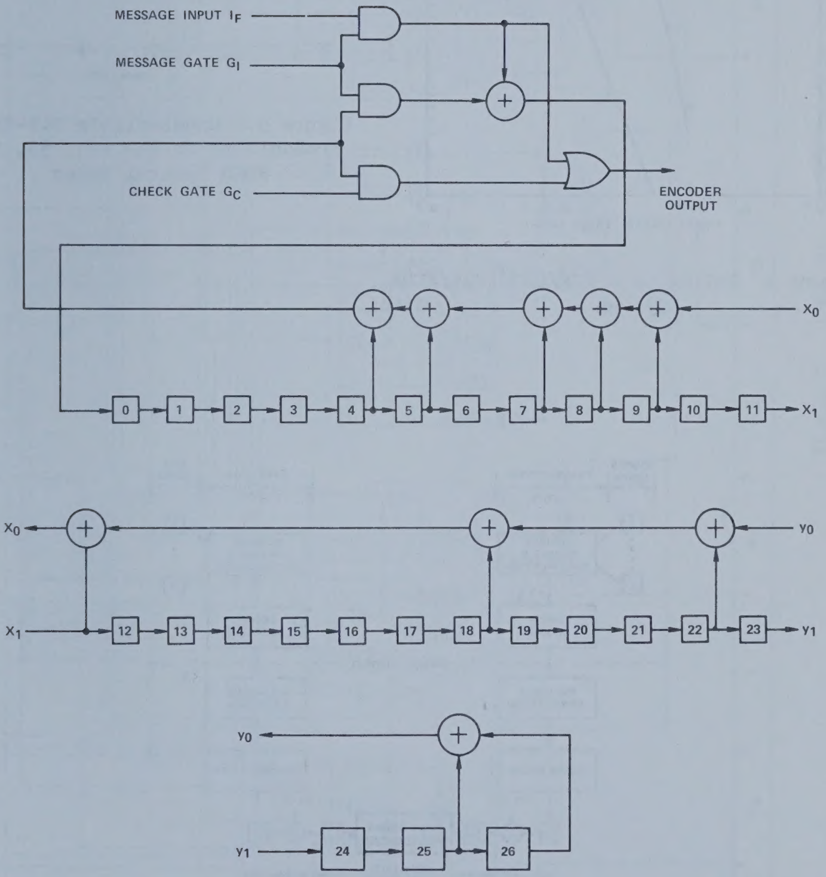

MESSAGE INPUT I_F
MESSAGE GATE G_I
CHECK GATE G_C
ENCODER OUTPUT
X_0
0
1
2
3
4
5
6
7
8
9
10
11
X_1
X_0
Y_0
X_1
12
13
14
15
16
17
18
19
20
21
22
23
Y_1
Y_0
Y_1
24
25
26

Figure 3. MARISAT Encoder

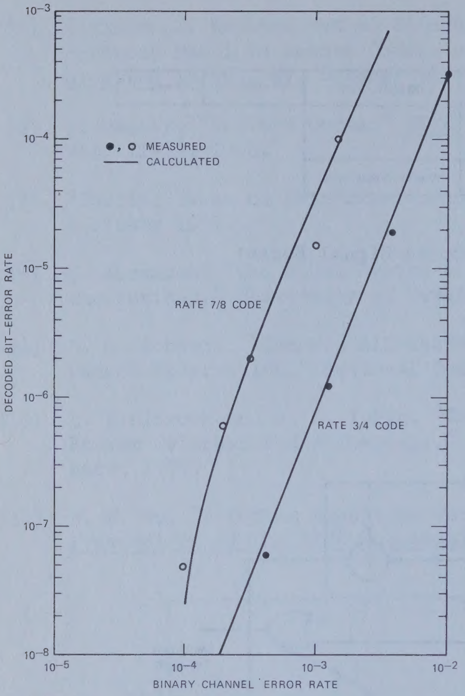


Figure 4. Coding in a PCM/CSOC/PSK/FDMA System

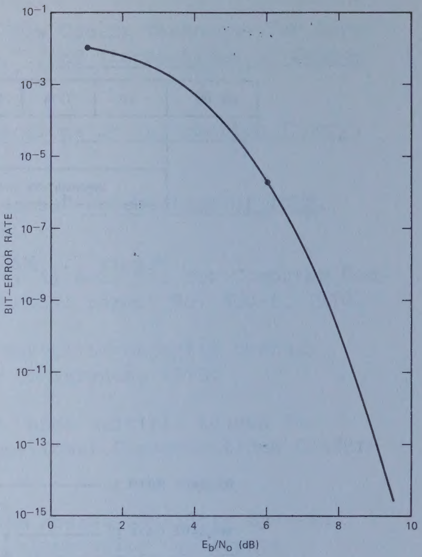


Figure 5. Undetectable Bit-Error Probability of the (73, 55, 5) FDMA Control Codec

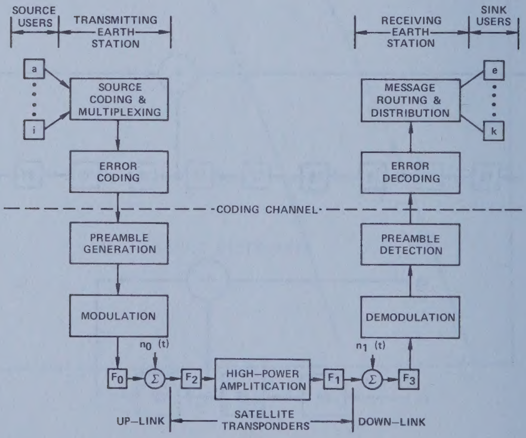
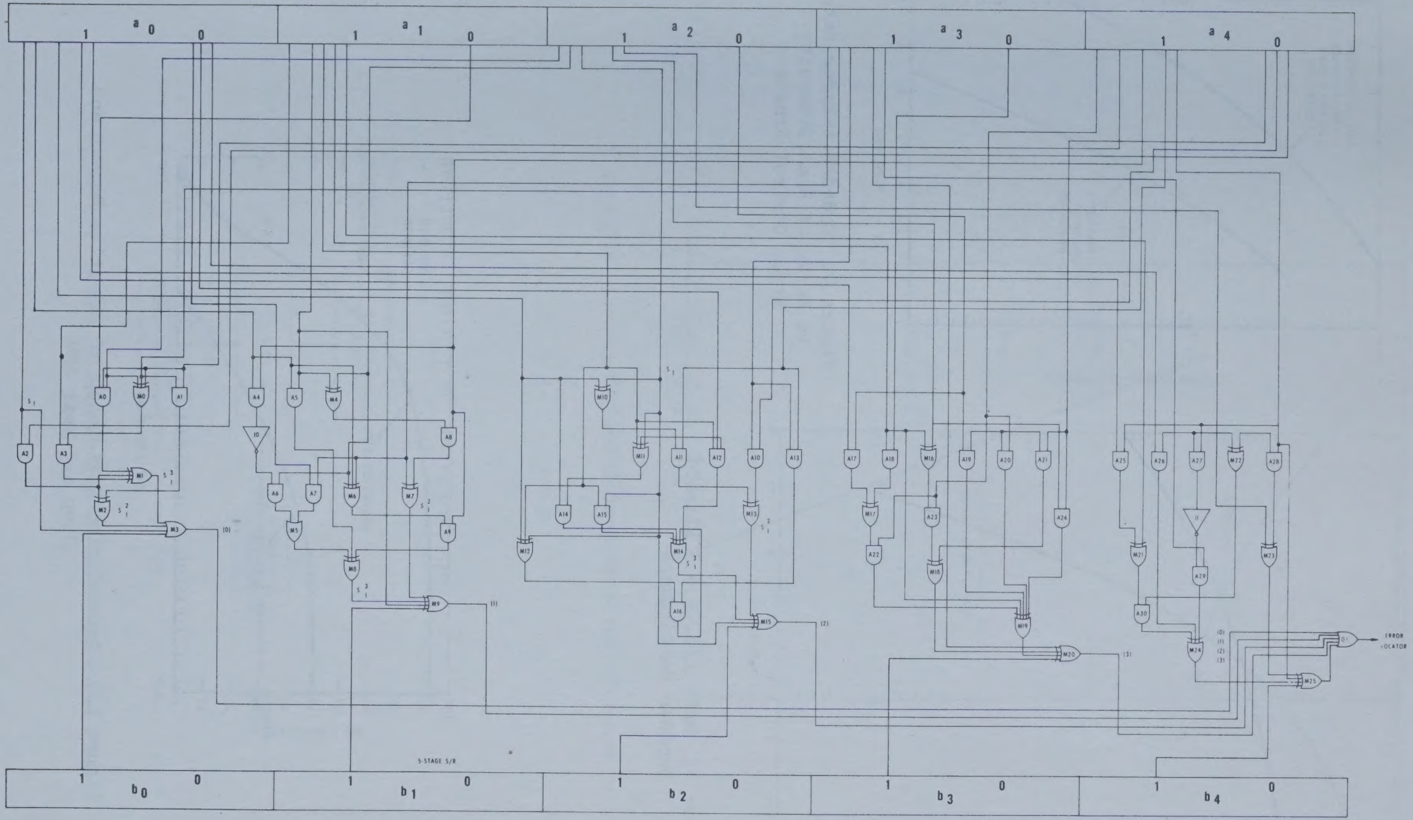

SOURCE USERS
TRANSMITTING EARTH STATION
RECEIVING EARTH STATION
SINK USERS
a
i
SOURCE CODING & MULTIPLEXING
MESSAGE ROUTING & DISTRIBUTION
e
k
ERROR CODING
ERROR DECODING
CODING CHANNEL
PREAMBLE GENERATION
PREAMBLE DETECTION
MODULATION
DEMODULATION
n0 (t)
n1 (t)
F0
Σ
F2
HIGH–POWER AMPLIFICATION
F1
Σ
F3
UP–LINK
SATELLITE TRANSPONDERS
DOWN–LINK

Figure 6. A TDMA Satellite Channel



1 a 0 0
1 a 1 0
1 a 2 0
1 a 3 0
1 a 4 0
S_1
S_1^3
S_1^2
(0)
(1)
(2)
(3)
ERROR LOCATOR
5-STAGE S/R
1 b 0 0
1 b 1 0
1 b 2 0
1 b 3 0
1 b 4 0

Figure 7. (31, 21, 5) BCH Decoder

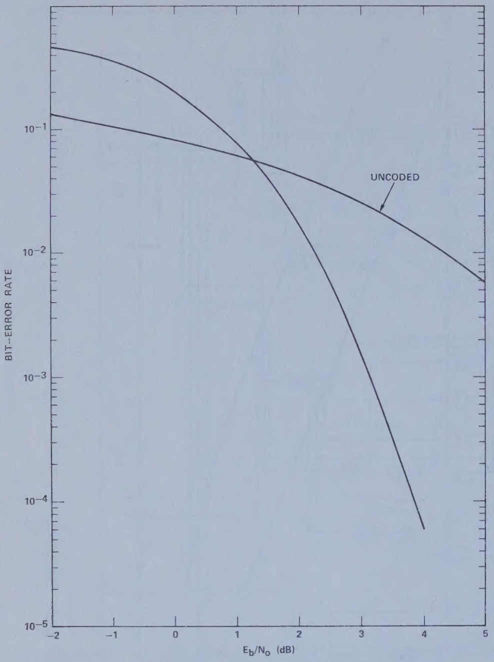


Figure 8. Rate 1/2, k = 7 Viterbi Decoder Performance

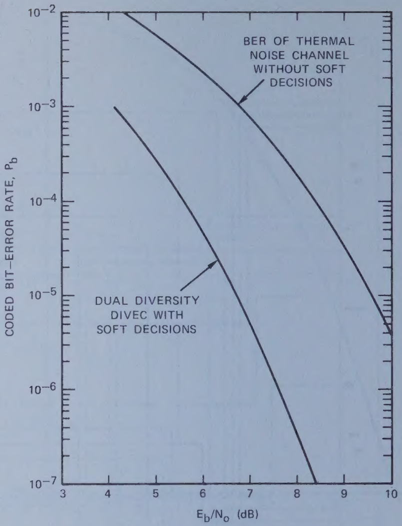


Figure 9. Coded Bit-Error Rate vs E_b/N_o of Dual Diversity Error Control Scheme

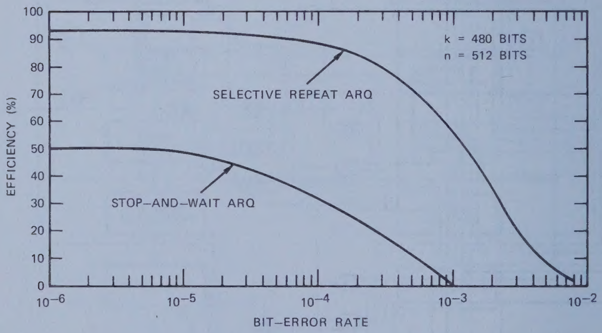


Figure 10. Transmission Efficiency of Selective Repeat and Stop and Wait ARQ

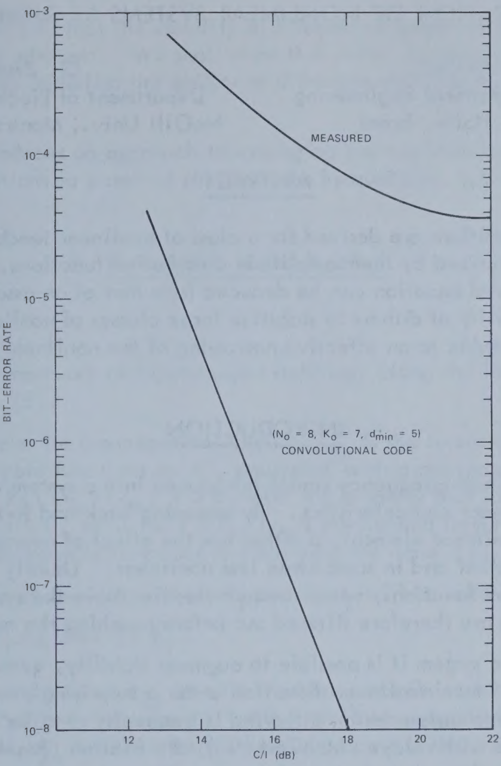


Figure 11. Coding for Multiple-Access Interference

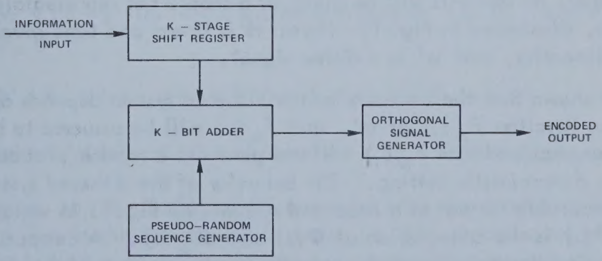

INFORMATION INPUT
K – STAGE SHIFT REGISTER
K – BIT ADDER
ORTHOGONAL SIGNAL GENERATOR
ENCODED OUTPUT
PSEUDO-RANDOM SEQUENCE GENERATOR

Figure 12. Multiple-Access Convolutional Encoding for Viterbi Decoding

ON THE STABILIZATION OF NONLINEAR SYSTEMS BY DITHER

N. SHNEYDOR
Department of Electrical Engineering
Technion, Haifa, Israel

G. ZAMES
Department of Electrical Engineering
McGill Univ., Montreal, Que., Canada

ABSTRACT

Stability conditions are derived for a class of nonlinear feedback equations. Dithers are characterized by their amplitude distribution functions. It is shown that stability of a dithered equation can be deduced from that of an associated smoothed equation. The ability of dithers to stabilize large classes of nonlinear systems is shown to be attributable to an effective narrowing of the nonlinear sector.

1. INTRODUCTION

A dither is a high-frequency signal introduced into a system with the object of modifying its nonlinear characteristics. By sweeping back and forth quickly across the domain of a nonlinear element, a dither has the effect of averaging the nonlinearity, making it smoother and in some sense less nonlinear. Usually dithers are periodic or stationary-random functions, whose frequencies lie above the system cut-off frequency, and which are therefore filtered out before reaching the output.

By dithering a system it is possible to augment stability, quench undesirable limit-cycles, and reduce nonlinear distortion under a surprisingly wide range of conditions. In servomechanism design dithering is frequently used for these purposes, for example, in systems with relays (MacColl [1]), dry friction (Besekerski [2]), and many others. From the point of view of theory, dither phenomena are interesting because they produce a self-linearizing tendency which appears to be a factor in a variety of nonlinear problems.

There have been many studies of dither by empyrical methods of the describing function and statistical-linearization type (ex., Pervozvanskii [3]), and there is a good understanding of the subject at an intuitive level. However, there seem to have been no rigorous analyses, especially in the area of feedback stability, and the precise manner in which dither augments stability has remained unclear.

In this paper, an analysis will be given of a simple but representative class of dithered systems, illustrated in Fig. 1. Here, H is linear and time invariant, Φ is a memoryless nonlinearity, and w' is a dither signal.

It will be shown that the behavior of the dithered system depends on the amplitude-distribution function $F_w(.)$ of w', and $F_w(.)$ will be assumed to be repetitive in time. It is emphasized that $F_w(.)$ will not generate a random process, but will be used in a purely deterministic setting. The behavior of the dithered system will be shown to be comparable to that of a smoothed system (see Fig. 1) in which the smoothed nonlinearity $\Phi^*(.)$ is the convolution of $\Phi(.)$ and $dF_w(.)$. A comparison test will be derived, relating the stability of the smoothed system to that of the dithered original whenever the repetition frequency is high enough.

It is well known [4] that the stability of a feedback system depends on the sector of its nonlinear element. We shall show that dither in effect narrows the sector and that this narrowing underlies the ability of dithers to stabilize large classes of systems.

This paper introduces an approach to averaging the nonlinearity, and a reduction of the feedback equations to a pair of simultaneous inequalities, which appear to be novel.

2. Formulation of the Problem

2.1 Preliminaries

We adopt the framework of input-output stability, along the lines of Ref. [4]. A good survey is Ref. [5].

Let $R\,(R^{+})$ denote the (nonnegative) Reals. L_{2e} is the linear space of R-valued, locally square-integrable functions on R^{+}, equipped with a one-parameter family of projection operators $P_T : L_{2e} \rightarrow L_{2e}$, $T \geq 0$, with the property that for $x \in L_{2e}$, $(P_T\, x)\,(t) = x\,(t)$ for $t \leq T$, or 0 for $t > T$. L_2 is the normed linear subspace of L_{2e} consisting of square-integrable functions and carrying the norm $\| x\,(.)\,\| = (\int_0^\infty x^2\, dt)^{1/2}$. The equivalent symbols $P_T\, x$ and x_T will be used interchangeably, and similarly for $\| P_T\, x \|$ and $\| x \|_T$.

An operator is any map of L_{2e} into itself which maps 0 into 0, and has the causality property, $P_T\, H = P_T\, H P_T$ for all $T \geq 0$. If H is an operator mapping a normed subspace A of L_{2e} into another normed subspace B, then H is AB-bounded if $\sup \{ \frac{\| Hx \|_B}{\| x \|_A} / x \in A,\, x \neq 0 \} < \infty$, and the supremum is the AB-induced norm of H, denoted by $g_{AB}\,(H)$. H is AB-Lipschitz if $\sup \{ \frac{\| Hx - Hy \|_B}{\| Hx - Hy \|_A} / x, y \in A,\, x \neq y \} < \infty$. The supremum is the AB-induced Lipschitz norm of H, denoted by $\overset{\circ}{g}_{AB}(H)$. Whenever $A = B$, only one symbol will be shown, for example $g_A(H) \triangleq g_{AA}(H)$. In the special case $A = B = L_2$, all norm subscripts will be omitted, i.e. $\| x \| \triangleq \| x \|_{L_2}$.

A function $\Psi : R \rightarrow R$ lies in the incremental sector $\{\alpha, \beta\}$, where $-\infty < \alpha \leq \beta < \infty$, if

$$\beta = \inf \{ \beta' \in R / \Psi\,(x) - \Psi(y) \leq \beta'\,(x - y),\ \forall\, x, y \in R,\ x \geq y \} \tag{1}$$

$$\alpha = \sup \{ \alpha' \in R / \Psi(x) - \Psi\,(y) \geq \alpha'\,(x - y),\ \forall\, x, y \in R,\ x \geq y \} \tag{2}$$

Ψ lies in the sector $\{\alpha, \beta\}$ if (1) and (2) are valid with the condition $\forall\ y \in R$ replaced by $y = 0$. If Ψ lies in the (incremental) sector $\{\alpha, \beta\}$, then the center $\theta \triangleq 1/2\,(\beta + \alpha)$ and radius $\rho = 1/2\,(\beta - \alpha)$ of the sector are constants determined by α and β. Ψ is called Lipschitz if it lies in an incremental sector $\{\alpha, \beta\}$, and the Lipschitz constant of Ψ is $\gamma = \max\,(\,|\,\alpha\,|,\ |\,\beta\,|\,)$.

2.2 The Dithered Feedback Equation

The system of Fig. 1a satisfies the feedback equation,

$$y' = H\Phi(u - y' + w') \tag{3}$$

in which,

1. u, y' and w' are elements of L_{2e} representing input, output, and dither respectively;
2. H is a convolution operator in L_{2e}, $(Hx)(t) = \int_0^\infty h(t-\tau)\, x(\tau)\, d\tau, \forall\, x \in L_{2e}$, $\forall\, t \geq 0$, with $h(.) \in L_1[0, \infty)$ and Fourier transform $\hat{H}(j\omega) = \int_0^\infty h(t) \exp(-j\omega t) dt$;
3. Φ is a (memoryless) operator in L_{2e}, determined by $(\Phi x)(t) = \varphi(x(t))$, where $\varphi(.): R \to R$ is a given function satisfying the conditions $\varphi(0) = 0$, and whose Lipschitz constant is γ.

It will be assumed that for each u and w' in L_{2e}, Eq. (3) has a unique solution for y' in L_{2e}.

2.3 The Unbiased Equation

The rest solution of Eq. (3), y'_o, is obtained when $u=0$, and is $y'_o = H\Phi(-y'_o + w')$. In general, $y'_o \neq 0$ and Eq. (3) will therefore be called biased. The rest solution can be viewed as a residual "ripple" attributable to the dither.

We are interested in the deviations $(y' - y'_o)$ of y' from rest, and therefore introduce the new variables

$y \triangleq y' - y'_o$ (the output deviation)

$w \triangleq w' - y'_o$ (the effective dither)

and the unbiased nonlinearity $\Phi_o : L_{2e} \to L_{2e}$, $\Phi_o(x) \triangleq \Phi(x+w) - \Phi(w)\ \forall\, x \in L_{2e}$. The new variables satisfy the following unbiased dithered feedback equation,

$$\begin{aligned} y &= H\,\Phi_o(u - y) \\ &= H\,\Phi(u - y + w) - H\,\Phi(w) \end{aligned} \tag{4}$$

Eqs. (3) and (4) are equivalent, i.e., if w' and y' in L_{2e} satisfy the former equation, then w and y in L_{2e} satisfy the latter, and vice-versa. The sector, center, radius, and Lipschitz constants of φ_o are identical to those of φ.

Henceforth, attention will be confined to the unbiased equation, Eq. (4), and so results will be obtained on the behaviour of Eq. (3) relative to its rest solution. The term "dither" will mean w. Once w is determined, w' can be found explicitly from the equation $w' = w + H\Phi w$.

Let $G_o : L_{2e} \to L_{2e}$ be the *closed loop operator* which maps each $u \in L_{2e}$ into the corresponding solution $y \in L_{2e}$ of Eq. (4). The object of this paper can now be stated: *To find conditions on the dither* w *which ensure that* G_o *is* L_2*-bounded.*

2.4 Amplitude Distribution Functions

Let $\mu(.)$ denote the length of a Lebesgue-measurable subset of R. Let $v(.) \in L_{2e}$, and (t_1, t_2) be any subinterval of R^+.

Definition: The *amplitude distribution function* (ADF) of $v(.)$ on (t_1, t_2) is the function $F_v : R \to [0, 1]$,

$$F_v(\xi) = \frac{\mu\{t / t \in (t_1, t_2),\ v(t) \le \xi\}}{(t_2 - t_1)} \quad .$$

The properties of ADF's are discussed in books on Probability (ex., Doob [6]). $F_v(\xi)$ is monotone non-decreasing in ξ, continuous on the right, $\lim_{\xi \to -\infty} F_v(\xi) = 0$, and $\lim_{\xi \to \infty} F_v(\xi) = 1$. Any function having these properties will be called an ADF.

Definition: v is *F-repetitive*, or F_v is *repetitive*, if there is a sequence $\{t_i\}$, $0 = t_o < t_1 < \dots$, unbounded from above, such that for $i = 1, 2 \dots$, the ADF of $v(.)$ on (t_{i-1}, t_i) equals the ADF of $v(.)$ on (t_o, t_1). The *maximal repetition interval*, is the supremum $\sup\{t_i - t_{i-1} / i = 1, 2, \dots\}$. The sequence $\{t_i\}$ will be called the F_v*-partition*.

Assumption: Henceforth the dither w of Eq. (4) will be a fixed, F-repetitive function in L_{2e}, with maximal repetition interval ΔT, and F_w-partition $\{t_i\}$.

2.5 The Smoothed Equation

Definition: For any $\Psi : R \to R$, the *smoothed image* $\Psi^* : R \to R$ is,

$$\Psi^*(\eta) = \int_{-\infty}^{\infty} \Psi(\xi)\, d_\xi F_w(\xi - \eta), \tag{5}$$

provided the Lebesgue-Stieltjes integral (5) exists for almost all η. Moreover, $\Psi_o^* : R \to R$ satisfies $\Psi_o^*(\eta) = \Psi^*(\eta) - \Psi^*(0)$.

Definition: The *smoothed images* of Φ and Φ_o are the operators Φ^* and Φ_o^* in L_{2e} satisfying the identities $(\Phi^* x)(t) = \varphi^*(x(t))$ and $(\Phi_o^* x)(t) = \varphi_o^*(x(t))$ for all $x \in L_{2e}$ and $t \in R^+$.

The *smoothed feedback equation* is,

$$\begin{aligned} y^* &= H \Phi_o (u - y^*) \\ &= H \Phi_o (u - y^*) - H \Phi^*(0), \end{aligned} \tag{6}$$

u and y^* being in L_{2e}.

It is shown in Appendix 1 that whenever φ lies in a (incremental) sector $\{a, b\}$, φ^* lies in a (incremental) sector $\{a^*, b^*\}$ and $a \le a^* \le b^* \le b$. In other words the *smoothed nonlinearity φ^* lies in a (incremental) sector not greater than that of the original φ.* By a suitable choice of dither distribution, the smoothed sector can often be made much smaller than the original.

2.6 High-Frequency Attenuation and the Sobolev Space S_{2p}

The high-frequency attenuation properties of H will be expressed by stipulating that its range be a Sobolev space (see Yosida [7]).

Definition: For any constant $p > 0$, the *Sobolev Space* S_{2p} is the linear subspace of L_2 containing every function $x(.)$ which is absolutely continuous on $[0, \infty)$, satisfies $x(0) = 0$, and whose derivative $\dot{x}(.)$ exists a.e. and is in L_2, and equipped[1] with the norm $\|x\|_{S_2} = (\|x\|^2 + \|p^{-1}\dot{x}\|^2)^{1/2}$.

If $x \in L_2$, a sufficient condition for $(Hx) \in S_{2p}$ is that $\sup_{\omega \in R} |\omega \hat{H}(j\omega)| < \infty$. For, then (Hx) has a derivative in L_2 by Parseval's Theorem, and $(Hx) \in L_2$ because H is L_2-bounded.

The abbreviation g_{LS} will be used for $g_{L_2 S_2}$.

2.7 Constant Terms

Various constants will be defined for use in the proofs.

$$\mathcal{K} = g_{s_2}(I - G_o^*) \qquad k_{LS} = \sup_{\omega \in R} |(1 + jp^{-1}\omega) H(j\omega)| \qquad p_2 = k_D \gamma/\sqrt{8}$$

$$k = \sup_{\omega \in R} |H(j\omega)| \qquad c = \mathcal{K} k_{LS} \gamma \qquad p_o = (\Delta T)_o^{-1}$$

$$k_D = \sup_{\omega \in R} |\omega H(j\omega)| \qquad p_1 = \mathcal{K} k_{LS} (\gamma + \gamma^*) p \Delta T/\sqrt{8}$$

$$(\Delta T)_o = \sup \{\Delta T / p_2 \Delta T \leq 1, \quad p_1 \Delta T + \frac{2 c p_2 \Delta T}{1 - p_2 \Delta T} \leq 1\}$$

3 Principal Results

Consider the feedback equation (4) and the associated smoothed equation (6). Recall that the dither w is F-repetitive with maximal interval ΔT.

Theorem 1 (A comparison test). If

i. $\sup_{\omega \in R} |\omega H(j\omega)| < \infty$, and

ii. G_o^* (the smoothed image of G_o) is a causal operator,

then, sufficient conditions for G_o to be L_2-bounded are that G_o^* be S_{2p}-bounded and $p_o \Delta T < 1$.

---

[1] The subscript-free symbol $\|\,.\,\|$ will be reserved for the L_2 norm.

Corollary 1.1 If $\sup_{\omega \in R} |\omega H(j\omega)| < \infty$ and $p_o \Delta T < 1$, then a sufficient condition for G_o to be L_2-bounded is that the smoothed system satisfy the circle criterion, i.e., for some constants a and b, $0 < a \le b < \infty$,

$$a \le \frac{\varphi(\xi)}{\xi} \le b, \qquad \forall \xi \in R, \qquad \xi \neq 0, \tag{7}$$

$$|\hat{H}(j\omega) + \frac{1}{2}(\frac{1}{a} + \frac{1}{b})| \le \frac{1}{2}(\frac{1}{a} - \frac{1}{b}), \ \forall \omega \in R, \tag{8}$$

and $\hat{H}(j\omega)$ does not encircle the $-\frac{1}{2}(\frac{1}{a} + \frac{1}{b})$ point.

The proofs of Theorem 1 and its Corollary are in Section 6.

Remarks Theorem 1 implies that for fast dithers the stability of the dithered system can be deduced from the S_2-boundedness of the smoothed one. If the smoothed system satisfies the circle criterion it is S_2-bounded, and so the original system must be L_2-bounded.

Stability in the bounded-input-bounded-output sense, or in the assymptotic sense, can be deduced from L_2-boundedness by standard methods (Ref. [5]).

The smoothed nonlinearity lies in a smaller sector than the original, and the stabilizing effect of dither can be viewed as caused by this effective reduction in sector.

4 Averages of φ^* and φ

The distribution of a sum, $x + w$, of independent variables is the convolution of their distributions, by a well known theorem of Probability (Doob [6]). A similar result will be obtained here for the special case where w is F-periodic and x is a step-function. It will follow that $\varphi(x + w)$ and $\varphi^*(x)$ have equal averages, and this fact will underlie our method of proof. In later sections, arbitrary elements of a Sobolev space will be approximated by step-functions.

Let (t_m, t_n) be a fixed, finite interval with end-points in the F_w-partition, and $\ell = t_m - t_n$.

4.1 Averages and Means

The average of any $v \in L_{2e}$ on (t_m, t_n), denoted by $\bar{v}$, is

$$\bar{v} \triangleq \frac{1}{\ell} \int_{t_m}^{t_n} v(t)\, dt \tag{9}$$

If $\Psi : R \to R$ is Lebesgue-Stjeltjes integrable with respect to F_v, the mean of Ψ w.r.t. F_v, denoted by $E_v(\Psi)$, is the Lebesgue-Stjeltjes integral $E_v(\Psi) \triangleq \int_{-\infty}^{\infty} \Psi(\eta)\, dF_v(\eta)$. By a theorem of Van-Vleck [8], if either $\overline{\Psi v}$ or $E_v(\Psi)$ exists, they are equal, i.e., $\overline{\Psi v} = E_v(\Psi)$.

4.2 Step Functions and ADF's of Sums

Definition: A step-function is any piecewise-constant function in L_{2e} continuous on the right. It will be assumed that every step-function x is synchronized with $w(.)$, i.e., the discontinuity points of $x(.)$ are F_w-partition points.

Lemma 1 If $x(.)$ is any step-function, and all ADF's are measured on (t_m, t_n), then (see Appendix 1 for proof), $F_{x+w}(\xi) = \int_{-\infty}^{\infty} F_w(\xi-\eta)\, dF_x(\eta)$.

Lemma 2 If $\Psi: R \to R$ is Borel-measurable, and x any step-function, then $\overline{\Psi(x+w)} = \overline{\Psi^*(x)}$ whenever either average exists on (t_m, t_n).

In particular, the conclusions of Lemma 2 hold when $\Psi(.)$ is the function $\varphi(.)$.

Proof: If $\overline{\Psi(x+w)}$ exists, then

$$\overline{\Psi(x+w)} = \int_{-\infty}^{\infty} \Psi(\xi)\, dF_{x+w}(\xi) = \int_{-\infty}^{\infty} \Psi(\xi)\, d_\xi \int_{-\infty}^{\infty} F_w(\xi-\eta)\, dF_x(\eta),$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Psi(\xi)\, d_\xi F_w(\xi-\eta)\, dF_x(\eta) = \int_{-\infty}^{\infty} \Psi^*(\eta)\, dF_x(\eta),$$

$$= \overline{\Psi^*(x)}.$$

Conversely, if $\overline{\Psi^*(x)}$ exists, the same arguments in reversed order prove Lemma 2.

As φ is Lipschitz it is Borel measurable, so it may be substituted for Ψ in Lemma 2. qed

5 S_{2p}-Approximations and Properties

In order to exploit the relation between the original φ and smoothed φ^* expressed by (13), L_2 functions will be approximated by step-functions. The approximations will converge for those functions having the smoothness properties associated with S_{2p}.

The F_w-partition is denoted by $\{t_i\}$.

Definition: The step-average $\tilde{x}(.) \in L_{2e}$ of $x(.) \in L_{2e}$ is the function $\tilde{x}(t) = (t_i - t_{i-1})^{-1} \int_{t_{i-1}}^{t_i} x(\tau)\, d\tau$, $\forall t \in [t_{i-1}, t_i)$, $i = 1, 2, \ldots$

$\tilde{x}(.)$ is a step-function, and the best L_2 approximation to $x(.)$ in the sense that if x_s is any other step-function (with discontinuity points in $\{t_i\}$ only), then $\|x_s - x\| \geq \|\tilde{x} - x\|$. This is proved by calculating the difference $\|x_s - x\|^2 - \|\tilde{x} - x\|^2$ and showing it to be equal to $\|x_s - \tilde{x}\|^2$ which is nonnegative. Another inequality we shall need is, $\|\tilde{x}\| \leq \|x\|$ as $\|x\|^2 - \|\tilde{x}\|^2 = \|x - \tilde{x}\|^2 \geq 0$. Similarly, $\|x - \tilde{x}\| \leq \|x\|$.

$(x-\tilde{x})$ is not necessarily small for small ΔT, but its integral is. In fact, if $D^{-1}: L_2 \to S_{2p}$ denotes the integration operator, $(D^{-1}x)(t) = \int_0^t x(\tau)\, d\tau$, the following Lemma, proved in Appendix 2, is obtained.

Lemma 3 For any $x(.) \in L_2$ and $T \in \{t_i\}$, $\| D^{-1}(x-\tilde{x}) \|_T \le \frac{\Delta T}{\sqrt{8}} \| x-\tilde{x} \|_T \le \frac{\Delta T}{\sqrt{8}} \| x \|_T$.

Let $D: S_{2p} \to L_2$ denote the differentiation operator, $Dx = \dot{x}$, $\dot{x}$ being the derivative of x a.e. Let $Q: S_{2p} \to L_{2e}$ be the mapping $Q = I + p^{-1} D$, I being the identity on S_{2p}. Then, by a simple calculation, for any $x \in S_{2p}$,

$$\|x\|_{S_{2p}} = \| Qx \|_{L_2}\,; \quad \| P_T x \|^2 + \| P_T \dot{x} \|^2 \le \| P_T Q x \|^2; \quad \| P_T x \| \le \| P_T Q x \|\,;$$

Lemma 4 For any $x(.) \in S_{2p}$ and $T \in \{t_i\}$,

$$\| x - \tilde{x} \|_T \le \frac{\Delta T}{\sqrt{8}} \| \dot{x} \|_T \le \frac{p\Delta T}{\sqrt{8}} \| Qx \|_T \le \frac{p\Delta T}{\sqrt{8}} \| x \|_{S_{2p}}\,.$$

In other words, the step-average approximation to x converges as $\Delta T \to 0$ provided $x \in S_{2p}$. The proof of Lemma 4 is in Appendix 2.

5.1 $L_2 - S_2$ Induced Norms

If $\sup_{\omega \in R} | \omega \hat{H}(j\omega) | < \infty$, the range of H is in S_{2p}, so the composition QH is well defined. Then, by Parseval's theorem and elementary Fourier transform properties, the $L_2 - S_2$ induced norm of H is

$$g_{LS}(H) = g(QH) = \sup_{\omega \in R} | (1 + p^{-1} j\omega) \hat{H}(j\omega) |$$

Let $Q^{-1}: L_2 \to S_{2p}$ be the inverse of Q. For any operator K it can be shown that

$$\overset{o}{g}_{S_2}(K) = \overset{o}{g}(QKQ^{-1})$$

Section 6 and Appendices. These sections contain most of the proofs. They have been omitted to save space, but may be obtained directly from G. Zames.

REFERENCES

[1] MacColl, L.A., Fundamentals of Servomechanisms, Van Nostrand, 1945, pp. 78 - 87.

[2] Bersekerskii, V.A., "Applying Vibrators to Eliminate Nonlinearities in Automatic Regulators", Automat. i Telemech., No. 8, 1947, pp. 411 - 417.

[3] Pervozvanskii, A.A., "An Approximate Method for the Investigation of Self-Oscillating Systems with Random Perturbations", Izv. Akad. Nauk, SSSR, OTN, No. 3, 1958, pp. 14 - 24.

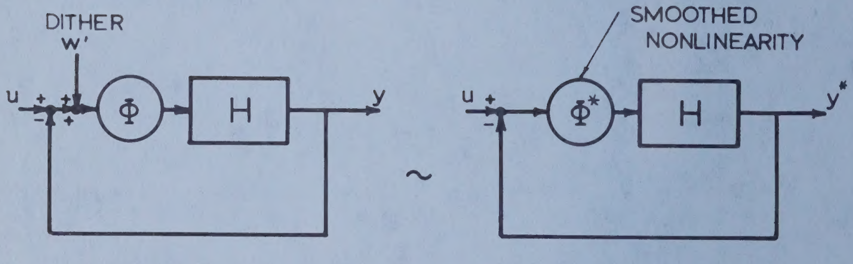
[4] Zames, G., "On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems", IEEE Trans. Automat. Control, Vol. AC-11, 1966, pp. 228 - 238, 465 - 476.

[5] Desoer, C.A. and Vidyasagar, M., Feedback Systems: Input-Output Properties, Academic Press, 1975.

[6] Doob, J.L., Stochastic Processes, Wiley, 1953.

[7] Yosida, K., Functional Analysis, Springer Verlag, 1968.

[8] Van Vleck, E.B., "Haskin's Momental Theorem and its Connection with Stjeltjes's Problem of Moments", Trans. Amer. Math. Soc., Vol. 18, 1917, pp. 326 - 330.



(a) (b)

FIG.1 A dithered feedback system.

L_2-STABILITY AND INSTABILITY OF LARGE SCALE SYSTEMS DESCRIBED BY INTEGRO-DIFFERENTIAL EQUATIONS

R. K. MILLER*
Department of Mathematics

Iowa State University
Ames, Iowa 50011

A. N. MICHEL**
Electrical Engineering Department and
Engineering Research Institute
Iowa State University
Ames, Iowa 50011

ABSTRACT

New results for (i) L_2-stability, (ii) L_2-instability, and (iii) asymptotic stability in the sense of Lyapunov for a large class of large-scale dynamical systems (also called composite systems, interconnected systems, and multiloop systems) described by Volterra integrodifferential equations are presented. These results, which allow usuage of frequency domain techniques, are applied in the stability analysis of the point kinetics model of a multi-core nuclear reactor.

I. INTRODUCTION

New results for L_2-stability and L_2-instability for a class of large-scale systems (also called interconnected systems, composite systems, multiloop systems, and the like) described by Volterra integrodifferential equations are established. It is shown that when the large-scale systems considered are bounded input bounded output stable on L_2 (BIBO stable on L_2), then they are also asymptotically stable in the sense of Lyapunov. The present results are proved for several forms of the describing equations, including the case of multiple-input multiple-output systems (MIMO systems) with interconnections. For the case of MIMO systems, conditions are established which guarantee instability when at least one subsystem is unstable.

In the present approach, the objective is to analyze large-scale systems in terms of lower-order subsystems and in terms of the system interconnecting structure. In order to demonstrate the usefulness of the proposed approach, the present results are applied to the linear point kinetics model of a multicore nuclear reactor.

The present BIBO stability results are motivated by those of Sandberg [20]-[21] and Zames [24]-[25] for single loop systems while the Lyapunov stability result is motivated by a result of Grossman and Miller [5]. The instability result constitutes a generalization of work by Vidyasagar [22]. Recent related results for BIBO stability of large-scale systems are due to Porter and Michel [17], [18] and Lasley and Michel [8]-[10]. For a general introduction to Volterra integral equations, refer to the monograph by Miller [11]. For additional specific information on resolvents which will be needed, refer to Miller and Grossman [5], [6] and Miller [12]. For a comprehensive summary of BIBO stability results, refer to [2] and [23].

---

* This author was supported by the National Science Foundation under Grant No. MPS 75-07622.

** This author was supported by the National Science Foundation under Grant ENG 75-14093 and by the Engineering Research Institute, Iowa State University.

II. NOTATION AND PRELIMINARIES

Let $C=[c_{ij}]$ denote an $m\times n$ matrix and let C^T be the transpose of C. If C and D are real $m\times n$ matrices, then $C\geq D$ means $c_{ij}\geq d_{ij}$ for all i and j. Let I denote the identity matrix, let $R^+=[0,\infty)$ and let R^m be the Euclidean m-space. If $x=[x_1,\text{---},x_m]^T\epsilon R^m$, then $|x|$ is the norm of x. Let L_p (or L_p^m if m needs to be emphasized) denote the set of all Lebesgue measurable functions $f:R^+\rightarrow R^m$ such that $\|f\|_p=[\int_0^\infty|f(t)|^p dt]^{1/p}$ is finite. When $p=2$, L_2 is a Hilbert space with inner product $\langle f,g\rangle=\int_0^\infty f(t)^T g(t)dt$ and $\|f\|_2=\langle f,f\rangle^{\frac{1}{2}}$. Given a subspace $M\subset L_2$, $M^\perp=\{f:\langle f,g\rangle=0 \text{ for all } g\epsilon M\}$. Also, let L_∞ (or L_∞^m if m needs to be emphasized) be the space of essentially bounded functions $f:R^+\rightarrow R^m$ and let $\|f\|_\infty=\operatorname{ess\,sup}_{t\geq 0}|f(t)|$. Let $C[0,\infty)$ be the set of all continuous functions $f:R^+\rightarrow R^m$ and $BC=\{f\epsilon C[0,\infty)\,|\,f(t)\rightarrow 0 \text{ as } t\rightarrow\infty\}$.

Given a function $f\epsilon L_p$, let f_T denote the truncation of f at time T. Also, let L_{pe} denote the extended space of L_p (see, e.g., Zames [24], [25]).

If $H:L_{2e}^k\rightarrow L_{2e}^\ell$, H is said to be L_2-stable if H maps L_2^k into L_2^ℓ. In this case the gain of H, g(H), is the smallest number K such that $\|(Hx)_T\|_2\leq K\|x_T\|_2$ for all $x\epsilon L_{2e}^k$ and all $T>0$. If the stable set of H, $S(H)=\{x\epsilon L_2^k: Hx\epsilon L_2^\ell\}$ is a proper subset of L_2^k, then H is said to be L_2-unstable, and the conditional gain of H, $g_c(H)$, is the smallest number K such that $\|(Hx)_T\|_2\leq K\|x_T\|_2$ for all $T>0$ and all x in the stable set S(H). H is interior conic (C,r) if $\|(Hx)_T-Cx_T\|_2\leq r\|x_T\|$ for some real constant $r\geq 0$ and some matrix C.

Let $\mathcal{L}_e$ denote the class of linear time-invariant operators on L_{pe} having the following properties: If $H\epsilon\mathcal{L}_e$, then there is a function $h\epsilon L_{1e}$ and two sequences $\{h_i\}$ and $\{t_i\}$, such that $t_i<t_{i+1}$ with $t_1=0$, $t_j\rightarrow\infty$ as $j\rightarrow\infty$, and

$$(Hx)(t)=\sum_{i=1}^\infty h_i x(t-t_i)+\int_0^t h(s)x(t-s)ds \tag{1}$$

for all $x\epsilon L_{pe}$. The class $\mathcal{L}$ will consist of all $H\epsilon\mathcal{L}_e$ such that the corresponding function h(t) and sequence $\{h_i\}$ satisfy the conditions

$$\sum_{i=1}^\infty |h_i|<\infty,\quad \int_0^\infty|h(t)|dt<\infty.$$

Let $H^*(s)$ and $h^*(s)$ denote the Laplace transforms of the operator H and of h(t), respectively (see Eq. (1)), where $s=\sigma+j\tau$, $j=\sqrt{-1}$ and $\sigma,\tau\epsilon R^1$. If $H\epsilon\mathcal{L}$, then the representation $H^*(s)$ is guaranteed to converge for all s with $\operatorname{Re}s=\sigma\geq 0$. Let $H^*(j\omega)$ denote the Fourier transform or frequency response of H. The resolvent of H, denoted by R, is the operator $R\epsilon\mathcal{L}_e$ given by

$$R^*(s)=(sI-H^*(s))^{-1}$$

Given $H\epsilon\mathcal{L}$, H is said to have property F when $\det(j\omega I-H^*(j\omega))\neq 0$ for all $\omega\epsilon R^1$. Moreover H will have property L if $\det(sI-H^*(s))\neq 0$ for all s such that $\operatorname{Re}s\geq 0$. It is known (see, e.g., Grossman and Miller [6] and Jordan and Wheeler [7]) that for $H\epsilon\mathcal{L}$ with resolvent R, H has property L if and only if $R\epsilon\mathcal{L}$. Also, H has property F if and only if there is a finite set $\{s_1,s_2,\text{---},s_N\}$ such that $\det(s_jI-H^*(s_j))=0$. For such an H one can find a set of matrices M_{jk} and an operator $S\epsilon\mathcal{L}$ such that

$$R^*(s)=S^*(s)+\sum_{j=1}^N\sum_{k=0}^{m_j}M_{jk}/(s-s_j)^k,\quad(\operatorname{Re}s_j>0). \tag{2}$$

The operator S is called the residual resolvent of H. (If R has property $\mathcal{L}$, then there are no such points s_j and so the resolvent and residual are identical.) The resolvent R and residual resolvent S are always of the special form

$$Rx(t) = \int_0^t r(t-u)x(u)du, \quad Sx(t) = \int_0^t s(t-u)x(u)du,$$

respectively. The matrix functions $r(t) = [r_{ij}(t)]$ and $s(t) = [s_{ij}(t)]$ are called the kernel of the operator R and S, respectively. When $H \in \mathcal{L}$ and has property L (or F) then $r_{ij} \in BC \cap L_2$ (or $s_{ij} \in BC \cap L_2$) for all indices i and j.

It is well known that the gain H of $\mathcal{L}$ on the space L_2 is

$$g(H) = \operatorname*{ess\ sup}_{-\infty < \omega < \infty} |H^*(j\omega)|.$$

If $H \in \mathcal{L}$ and has property L then by the same reasoning the gain of the resolvent is

$$g(R) = \operatorname*{ess\ sup}_{-\infty < \omega < \infty} |(I - H(j\omega))^{-1}|$$

A regular operator H is an element of $\mathcal{L}_e$ such that for the corresponding sequence $\{h_i\}$, $h_i = 0$ for all $i > 1$. If H is regular, the resolvent kernel $r(t) = [r_{ij}(t)]$ has continuously differentiable elements $r_{ij}(t)$ on R^+ (see Grossman and Miller [6]). Also $r'_{ij} \in L_1 \cap L_2 \cap BC$ when H and $R \in \mathcal{L}$.

A linear integrodifferential equation with kernel $H \in \mathcal{L}_e$ and forcing function $f \in L_{2e}$ is the equation

$$y'(t) = Hy(t) + f(t), \quad t \geq 0 \tag{L}$$

with initial condition $y(0) = y_0$. A solution y(t) of (L) is a function which is absolutely continuous on any finite interval $0 \leq t \leq T$ with derivative $y'(t)$ at almost all points $t \in R^+$. If $H \in \mathcal{L}_e$ and $f \in L_{2e}$ it is well known that (L) has a unique solution. Indeed, if r(t) is the kernel of the resolvent R, then

$$y(t) = r(t)y_0 + \int_0^t r(t-s)f(s)ds,$$

or $y = R[\delta x_0 + f]$, where δ is the delta function.

All nonlinear integrodifferential equations considered herein are of the form

$$x'(t) = Hx(t) + B(t,x_t) + f(t), \quad t \geq 0 \tag{N}$$

with $x(0) = x_0$. Here $H \in \mathcal{L}_e$, $f \in L_{2e}$ and B is a continuous functional. These assumptions insure that (N) has a solution (which will be unique if the functional B satisfies the usual Lipschitz condition). If one thinks of $B(t,x_t) + f(t)$ as known, then from (3) the following variation of constants formula follows:

$$x = R[\delta x_0 + f] + R[B(t,x_t)]. \tag{V}$$

Remark 1. Let r(t) be the resolvent kernel of the regular resolvent R. If B has the regular form

$$B(t,x) = B_1(t,x(t)) + \int_0^t h(t-s)B_2(s,x(s))ds, \tag{4}$$

then (V) can be put into an alternate form. Since

$$r'(t) = r(t)h_1 + \int_0^t r(t-s)h(s)ds = r(t)h_1 + r*h(t),$$

where * denotes convolution, then $r*h = r' - rh_1$. The last term in (V) is of form

$$R[B_1 + h*B_2] = R(B_1) + r*h*B_2 = R(B_1) + (r' - rh_1)*B_2$$
$$= R(B_1 - h_1B_2) + r'*B_2.$$

Thus, (V) can be written in the alternate form

$$x = R[\delta x_0 + f] + R[B_1(t,x(t)) - h_1B_2(t,x(t))] + r'*B_2(t,x(t)). \quad (V')$$

Equation (N) is called *L_2-stable* (i.e., BIBO stable on L_2) if for each $f \in L_2$ and initial conditions $x_0 \in R^n$, all solutions of (N) are in L_2. Otherwise (N) is called *L_2-unstable*. System (N) is called *stable in the sense of Lyapunov* if for any $\varepsilon > 0$ there is a $\delta > 0$ such that when $|x_0| \le \delta$ and $\|f\|_2 \le \delta$ then $|x(t)| < \varepsilon$ for all $t \ge 0$. It is *asymptotically stable in the sense of Lyapunov* if in addition there is a $\delta_0 > 0$ such that $x(t) \in BC$ when $|x_0| \le \delta_0$ and $\|f\|_2 \le \delta_0$.

III. MAIN RESULTS

Systems are considered which can be described by the set of nonlinear integrodifferential equations

$$x_k'(t) = \sum_{j=1}^{N} [H_{kj}x_j(t) + B_{kj}(t,x_{jt})] + f_k(t) \quad (5)$$

for $k = 1,2,---,N$ and $t \ge 0$, with initial conditions $x_k(0) = x_{0k}$. Here $x_k: R^+ \to R^{n_k}$, $f_k \in L_2^{n_k}$, $H_{kj}: L_{2e}^{n_j} \to L_{2e}^{n_k}$ is in $\mathcal{L}_e$, and B_{ij} is a continuous nonlinear functional. Letting $x = [x_1, x_2, ---, x_N]^T$, $f = [f_1, f_2, ---, f_N]^T$, $B = [\Sigma B_{1j}, --- ---, \Sigma B_{Nj}]^T$ and $H = [H_{jk}]$, system (5) assumes the form (N) with initial condition $x_0 = [x_{10}, ---, x_{N0}]^T$ and $m = \sum_{j=1}^{N} n_j$. The *matrix of gains* $g(B)$ is defined as $g(B) = [g(B_{ij})]$. The matrix of gains $g(C)$ of any other matrix C is defined similarly.

When (N) is of the form (5), one speaks of a *large-scale system* (N) *with decomposition* (5).

Theorem 1. If $g(B_{kj}) < \infty$ for all k and j, if $H = [H_{ij}] \in \mathcal{L}$, if H has property L with resolvent $R = [R_{ij}]$, and if the successive principal minors of the test matrix $I - g(R)g(B)$ are all positive, then (5) is L_2-stable.

Theorem 2. Let B_{ij} be regular in the sense of (4), let $H = [H_{kj}] \in \mathcal{L}$ be regular, satisfy property L, have resolvent $R = [R_{kj}]$ with kernel $[r_{kj}]$, let T_{kj} be the operator defined by $T_{kj}x = r_{kj}' * x$, and let $T = [T_{kj}]$. If the successive principal minors of the test matrix $I - g(R)g(B_1 - h_1B_2) - g(T)g(B_2)$ are all positive, then (5) is L_2-stable.

Theorem 3. If the hypotheses of Theorem 1 or Theorem 2 are true, then (5) is asymptotically stable in the sense of Lyapunov.

Remark 2. In some applications of Theorem 2, $B_1 = h_1B_2$. In this case $g(B_1 - h_1B_2)$ is zero.

Remark 3. Consider the system of equations

$$x_k'(t) = f_k(t) + \sum_{j=1}^{N}[G_{kj}(t,x_j(t)) + \int_0^t M_{kj}(t-s)N_{kj}(s,x_j(s))ds]$$

where G_{kj} is interior conic (m_{kj},r_{kj}) and N_{kj} is interior conic (n_{kj},w_{kj}). In this case one can rewrite the equation for $x_k'(t)$ as

$$x_k'(t) = f_k(t) + \sum_{j=1}^{N} m_{kj}x_j(t) + \int_0^t M_{kj}(t-s)n_{kj}x_j(s)ds$$

$$+ \sum_{j=1}^{N}[G_{kj}(t,x_j(t)) - m_{kj}x_j] + \int_0^t M_{kj}(t-s)[N_{kj}(s,x_j(s))$$

$$- n_{kj}x_j(s)]ds.$$

This equation has the correct form for Theorem 2 since $B_1 = [G_{kj}(t,x_j(t)) - m_{kj}x_j]$, $g(B_1) \le [r_{kj}]$, $B_2 = [M_{kj} * (N_{kj}(t,x_j) - n_{kj}x_j)]$, $g(B_2) \le g([M_{kj}])[w_{kj}]$, $h_1 = [m_{kj}]$, and $h(t) = [M_{kj}(t)n_{kj}]$.

Of special interest in applications are systems described by (5) with $H_{ij} = 0$ for all $i \ne j$, i.e., MIMO systems of the form

$$x_k'(t) = H_k x_k(t) + \sum_{j=1}^{N} B_{kj}(t,x_{jt}) + f_k(t), \tag{6}$$

$k = 1,\text{---},N$, with $x_k(0) = x_{k0}$. This equation can be viewed as an interconnection of N isolated subsystems of the form

$$S_k\colon\ z_k'(t) = H_k z_k(t) + f_k(t), \tag{7}$$

$z_k(0) = z_{k0}$, $k = 1,\text{---},N$, with interconnecting structure specified by B_{kj}, $k,j = 1,\text{---},N$.

Corollary 1. In (6) assume that

(i) $H_k \in \mathcal{L}$ with stable resolvent R_k, $k = 1,\text{---},N$;
(ii) $g(B_{kj}) < \infty$, $k, j = 1,\text{---},N$;
(iii) $f_k \in L_2$, $k = 1,\text{---},N$; and
(iv) the successive principal minors of the test matrix $M = [m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 - g(R_i)g(B_{ij}), & i = j \\ -g(R_i)g(B_{ij}), & i \ne j \end{cases}$$

are all positive.

Then system (6) is L_2-stable and also asymptotically stable in the sense of Lyapunov.

Under certain assumptions one can show that if at least one of the isolated systems (7) is L_2-unstable, then the interconnected system (6) is also L_2-unstable.

Theorem 4. Assume that for system (6) the following conditions hold:

(i) $H_k \in \mathcal{L}$, $k = 1,\text{---},N$;
(ii) $g(B_{ij}) < \infty$, $i, j = 1,\text{---},N$;
(iii) $f_j \in L_2$, $j = 1,\text{---},N$;
(iv) for each k, H_k satisfies either property F and/or property L and for at least one k_0 it satisfies property F only; and
(v) Let $M = [m_{ij}] = [g(B_{ij})g_c(R_j)]$, where $g_c(R_i) = g(R_i)$ when R_i is stable, and let $\rho(M)$ be the spectral radius of M. Assume that either $\rho(M) < 1$ or $\rho(M) = 1$ and $m_{ij} > 0$ for all $i, j = 1,\text{---},N$.

Then system (6) is L_2-unstable.

Theorem 4 is a corollary of Theorem 5 given below. This result yields sufficient conditions for L_2-instability of interconnected systems described by equations of the form

$$\left.\begin{aligned} e_k &= f_k - \sum_{j=1}^{m} B_{kj}x_j \\ x_k &= R_k(e_j + v_j), \end{aligned}\right\} \tag{8}$$

$k = 1,---,N$, where $f_k, v_k \epsilon L_2^{n_k}$, $e_k, x_k \epsilon L_{2e}^{n_k}$, $R_k \epsilon \mathcal{L}_e^{n_k}$, and $B_{kj}: L_{2e}^{n_j} \to L_{2e}^{n_k}$. Define $M_k = \{x \epsilon L_{2e}^{n_k}: R_k x \epsilon L_2^{n_k}\}$.

Theorem 5. Suppose $M_k^{\perp} \neq \{0\}$ for at least one $k = k_0$ and $g(B_{kj}) < \infty$ for all k and j. Let $M = [m_{ij}] = [g(B_{ij})g_c(R_j)]$ and let $\rho(M)$ be the spectral radius of M. Assume that either $\rho(M) < 1$ or $\rho(M) = 1$ and $m_{ij} > 0$ for all i and j. If $f_k + v_k \epsilon M_k^{\perp}$ for all k and $f_k + v_k \not\equiv 0$ when $k = k_0$, then system (8) is L_2-unstable, i.e., $x_j \not\in L_2$ for at least one value j.

Remark 4. For BIBO stability results of system (8) refer to Porter and Michel [17], [18] and Lasley and Michel [9]. Theorem 5 may be viewed as the instability counterpart of the results of [9], [17], [18]. Theorem 5 is a generalization of a result reported in [22]. Specifically, in [22] it is required that all subsystems be L_2-unstable and all $v_k = 0$.

IV. AN EXAMPLE

The present results are now applied to study the stability properties of a coupled core nuclear reactor. For a general discussion of reactor models, refer to [1]. The point kinetics model for a reactor with M cores, as described in [16], is given by

$$\left.\begin{aligned} p_j'(t) &= \frac{\rho_j - \epsilon_j - \beta_j}{\Lambda_j} p_j(t) + \frac{\rho_j}{\Lambda_j} + \sum_{i=1}^{6} \frac{\beta_{ij}}{\Lambda_j} c_{ij}(t) \\ &\quad + \frac{1}{\Lambda_j} \sum_{k=1}^{M} \epsilon_{jk} \frac{P_{k0}}{P_{j0}} \int_0^t h_{kj}(t-s)p_k(s)ds \\ c_{ij}'(t) &= \lambda_{ij}[p_j(t) - c_{ij}(t)] \end{aligned}\right\} \tag{9}$$

for $j = 1,---,M$ and $i = 1,---,6$. Here $p_j(t) = [P_j(t) - P_{j0}]/P_{j0}$ and $c_{ij}(t) = (\lambda_{ij}\Lambda_j/P_{j0}\beta_{ij})(C_{ij}(t) - C_{ij0})$, where P_j denotes the power in the j^{th} core, C_{ij} represents the effective concentration of the i^{th} precursor in the j^{th} core, β_{ij}, λ_{ij}, ϵ_j, ϵ_{kj}, Λ_j are positive constants, $\beta_j = \sum_{i=1}^{6}\beta_{ij}$, and $h_{kj}(t)$ is the coupling function relating neutron migration from the k^{th} to the j^{th} core. P_{j0} and C_{ij0} are the equilibrium power and precursor concentrations in the j^{th} core while ρ_j is the reactivity in the j^{th} core. Assume that

$$\rho_j(t) = \int_0^t W_j(t-s)p_j(s)ds$$

is correct at least to linear terms, where the feedback function W_j is in L_1.

Solving for $c_{ij}(t)$ in terms of $p_j(t)$, one has

$$c_{ij}(t) = c_{ij}(0)e^{-\lambda_{ij}t} + \lambda_{ij}(e^{\lambda_{ij}t} * p_j).$$

Substituting $c_{ij}(t)$ into the equation for $p_j'(t)$ and linearizing, one obtains

$$p_j'(t) = f_j(t) - \frac{(\epsilon_j + \beta_j)}{\Lambda_j} p_j + \frac{W_j * p_j}{\Lambda_j} + \left(\sum_{i=1}^{6} \frac{\beta_j \lambda_{ij}}{\Lambda_j} e^{-\lambda_{ij} t} \right) * p_j$$

$$+ \frac{1}{\Lambda_j} \sum_{k=1}^{M} \epsilon_{kj} \frac{P_{k0}}{P_{j0}} h_{kj} * p_k, \tag{10}$$

with $p_j(0)$, $j = 1,---,M$ given.

Corollary 1 is now applied, with $x_j = p_j$ and $n_j = 1$ for all j, with

$$(H_j x)(t) = -\left[\frac{\epsilon_j + \beta_j}{\Lambda_j}\right] x(t) + \frac{1}{\Lambda_j} \int_0^t \left[W_j(t-s) + \sum_{i=1}^{6} \beta_{ij} \lambda_{ij} e^{-\lambda_{ij}(t-s)}\right] x(s)\, ds,$$

$$f_j(t) = \sum_{i=1}^{6} c_{ij}(0) e^{-\lambda_{ij} t},$$

and

$$B_{jk}(t, x_t) = \frac{1}{\Lambda_j} \frac{P_{k0}}{P_{j0}} (h_{kj} * x)(t).$$

The resolvent R_j has Laplace transform $R_j^*(s) = 1/D_j^*(s)$ and is stable if and only if

$$D_j(s) \equiv s + \left(\frac{\epsilon_j + \beta_j}{\Lambda_j}\right) + \sum_{i=1}^{6} \frac{\beta_{ij} \lambda_{ij}}{\Lambda_j (s + \lambda_{ij})} + W_j^*(s) \neq 0$$

in the half plane $\mathrm{Re}\, s \geq 0$. This condition can be checked graphically by plotting the Fourier transform of $D_j(s)$. Moreover, $1/g(R_j)$ is equal to the minimum distance from the graph of $D_j(j\omega)$, $-\infty < \omega < \infty$, to the origin in the complex plane. If the successive principal minors of the test matrix $M = I - [g(R_i) g(B_{ij})]$ are all positive, then for all initial values $p_j(0)$ and $c_{ij}(0)$ the solutions of (9) are in $L_2 \cap BC$. Moreover, they are stable in the sense that given $\epsilon > 0$ there exists $\delta > 0$ such that when $|p_j(0)| \leq \delta$ and $|c_{ij}(0)| \leq \delta$ for all i and j, then $|p_j(t)| \leq \epsilon$ and $|c_{ij}(t)| \leq \epsilon$ for all i and j and for all $t \geq 0$.

If $D_j(j\omega) \neq 0$ for $-\infty < \omega < \infty$ and $j = 1,---,M$ but $D_j(s_0) = 0$ for some s_0, $\mathrm{Re}\, s_0 > 0$ and some j, and if the matrix $M = [g(B_{ij}) g_c(R_j)]$ has spectral radius less than one (e.g., weak interconnections between cores), then Theorem 4 implies that (9) is L_2-unstable. Since for this type of linear systems, L_2-stability and asymptotic stability in the sense of Lyapunov are equivalent (see [12]), this implies that (10) is Lyapunov unstable and hence so is (7). The results in [14] imply the fact that when a linearized equation of the form (10) are Lyapunov unstable, this instability will carry over to the corresponding unlinearized equations (9). Thus, the solutions of (9) are unstable.

V. PROOF OF MAIN RESULTS

Presently the results of Section III are proved.

Proof of Theorem 1. The variation of constants formula (V) applies. Thus,

$$x(t) = r(t)x(0) + Rf(t) + RB(t, x_t) \tag{11}$$

on $0 \leq t < \infty$. For any $T > 0$ if $\|x\|_T = \|x_T\|_2$ and $\|x(0)\| = [\,|x_1(0)|, ---, |x_m(0)|\,]^T$, then

$$\|x\|_T \le [\|r_{ij}\|_2]\|x(0)\| + g(R)\|f\|_2 + g(R)g(B)\|x\|_T,$$

or

$$[I - g(R)g(B)]\|x\|_T \le [\|r_{ij}\|_2]\|x(0)\| + g(R)\|f\|_2.$$

Since the successive principal minors of the test matrix $I - g(R)g(B)$ are all positive, this test matrix has an inverse ρ whose entries are all nonnegative (see [3], [15]). Thus,

$$\|x\|_T \le \rho([\|r_{ij}\|_2]\|x(0)\| + g(R)\|f\|_2). \tag{12}$$

Since (12) is true for all $T > 0$, the proof is complete.

Proof of Theorem 2. The variation of constants formula (V') applies. Thus,

$$x(t) = r(t)x(0) + Rf(t) + R[B_1(t,x(t)) - h_1B_2(t,x(t))] + r' * B_2(t,x(t))$$

on R^+. For any $T > 0$,

$$\|x\|_T \le [\|r_{ij}\|_2]\|x(0)\| + g(R)\|f\|_2 + g(R)g(B_1 - hB_2)\|x\|_T + g(r')g(B_2)\|x\|_T$$

or

$$[I - g(R)g(B_1 - h_1B_2) - g(r')g(B_2)]\|x\|_T \le [\|r_{ij}\|_2]\|x(0)\| + g(R)\|f\|_2.$$

The assumed condition on the test matrix implies (as in the proof of Theorem 1) that it has an inverse ρ whose entries are all nonnegative. Thus,

$$\|x\|_T \le \rho([\|r_{ij}\|_2]\|x(0)\| + g(R)\|f\|_2)$$

for all $T > 0$. This completes the proof.

Proof of Theorem 3. The proofs under the hypotheses of Theorem 1 and of Theorem 2 are essentially the same, so that only one case will be considered. Assuming the hypotheses of Theorem 1, it will be shown that (5) is asymptotically stable in the sense of Lyapunov.

Note that if $s, \varphi \epsilon L_2$, then the convolution $s * \varphi \epsilon L_\infty$ and indeed by the Schwarz inequality, $|(s * \varphi)(t)| \le \|s\|_2\|\varphi\|_2$. Moreover, $r * \varphi \epsilon BC$ (see for example Rudin [19, p. 4]). Since R is L_2-stable, the kernel $r \epsilon L_2 \cap BC$ and (11) implies

$$|x(t)| \le [\|r_{ij}\|_\infty]\|x(0)\| + [\|r_{ij}\|_2]\|f\|_2 + [\|r_{ij}\|_2]g(B)\|x\|_2.$$

The last in equality and (12) imply that

$$|x(t)| \le [\|r_{ij}\|_\infty]\|x(0)\| + [\|r_{ij}\|_2]\|f\|_2$$

$$+ [\|r_{ij}\|_2]g(B)\rho([\|r_{ij}\|_2]\|x(0)\| + g(R)\|f\|_2).$$

This proves the Lyapunov stability.

To show that $x \epsilon BC$, use (11). Since $r \epsilon BC$ and since $s * \varphi \epsilon BC$ whenever $s, \varphi \epsilon L_2$, (11) implies that $x \epsilon BC$ for all $x(0) \epsilon R^m$ and all $f \epsilon L_2$.

Proof of Theorem 4. Equations (6) can be written in the equivalent form

$$e_k(t) = f_k(t) + \sum_{j=1}^{N} B_{kj}(t, x_{jt}) \tag{13}$$

$$x_k'(t) = H_k x_k(t) + e_k(t), \tag{14}$$

$x_k(0) = x_{k0}$, $k = 1, ---, N$. Chooseing $x_{k0} = 0$ for all k and integrating (14), one obtains $x_k = R_k e_k$. This equation along with (13) are in the proper form for Theorem 5 to apply.

Proof of Theorem 5. The proof is similar to the proof of Theorem 3 in [22]. For purposes of contradiction, assume that $x_k \epsilon L_2$ for all k. If $\rho(M) < 1$, then one can replace each value $g_c(H_i)$ by $g_c(H_i) + A$ and each value $g(B_{ij})$ by $g(B_{ij}) + A$, where A is small and positive. If A is sufficiently small, the new matrix M will still have $\rho(M) < 1$ but now all entries of $[m_{ij}]$ are positive. Assume that this replacement has been accomplished. Then $m_{ij} > 0$ for all i and j and by the Perron Theorem (see, e.g., [4, p. 53]), there exist numbers $\lambda_j > 0$ such that $\Lambda = [\lambda_1, ---, \lambda_m]$ is a row eigenvector corresponding to the dominant eigenvalue $\rho(M)$ of M.

Put $z_i = \sum_{j=1}^{m} B_{ij} x_j = \sum_{j=1}^{m} B_{ij} H_j (e_j + v_j)$. Then $z_i = f_i + v_i - (e_i + v_i)$. Since $(f_i + v_i) \epsilon M_i^{\perp}$ and $(e_i + v_i) \epsilon M_i$, it follows that $z_i \epsilon L_2$ and

$$\|z_i\|_2^2 = \|f_i + v_i\|_2^2 + \|e_i + v_i\|_2^2 \geq \|e_i + v_i\|_2^2.$$

Since $\|f_i + v_i\| > 0$ for at least one $i = k_0$ and since all λ_i are positive, it follows that

$$\sum_{i=1}^{m} \lambda_i \|z_i\|_2 > \sum_{i=1}^{m} \lambda_i \|e_i + v_i\|_2. \tag{15}$$

The definition of z_i and the assumptions given yield the estimate

$$\|z_i\|_2 \leq \sum_{j=1}^{m} \|B_{ij} x_j\|_2 \leq \sum_{j=1}^{m} g(B_{ij}) \|x_j\|_2$$

$$\leq \sum_{j=1}^{m} g(B_{ij}) g_c(H_j) \|e_j + v_j\|_2 = \sum_{j=1}^{m} m_{ij} \|e_j + v_j\|_2. \tag{16}$$

Since Λ is a row eigenvector of M it follows that

$$\sum_{i=1}^{m} \lambda_i m_{ij} = \rho(M) \lambda_j$$

for all j. Thus, (16) and $\rho(M) \leq 1$ imply that

$$\sum_{j=1}^{m} \lambda_i \|z_i\|_2 \leq \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i m_{ij} \|e_j + v_j\|_2 =$$

$$= \sum_{j=1}^{m} \rho(M) \lambda_j \|e_j + v_j\|_2 \leq \sum_{j=1}^{m} \lambda_j \|e_j + v_j\|_2.$$

This contradicts (15) and the theorem is proved.

VI. CONCLUDING REMARKS

At this point some additional comments are in order.

1. It is emphasized that the present results yield not only L_2-stability and L_2-instability conditions for systems described by integrodifferential equations, but also asymptotic stability conditions in the sense of Lyapunov.

2. Some (but not all) of the results presented offer the advantage of allowing stability analysis of complex systems in terms of lower order subsystems and in terms of system interconnecting structure.

3. It is possible to analyze large classes of problems by the present results, using frequency domain techniques (with all the practical advantages which such techniques offer).

4. The purpose of the application of Section IV is to present a noncontrived example. No claims of a significant contribution in the area of nuclear reactors are intended.

REFERENCES

[1] Z. Akcasu, G. S. Lillouche and L. M. Shotkin, Mathematical Methods in Nuclear Reactor Dynamics, Academic Press, N. Y., 1971.

[2] C. A. Desoer and M. Vidyasagar, Feedback Systems: Input-Output Properties, Academic Press, N. Y., 1975.

[3] M. Fiedler and V. Ptak, "On Matrices with Non-Positive Off-Diagonal Elements and Positive Principle Minors", Czech. Math. J., Vol. 12, 1962, pp. 382-400.

[4] F. R. Gantmacher, Matrix Theory, Vol. II, Chelsea Publishers, N.Y., 1959.

[5] S. I. Grossman and R. K. Miller, "Perturbation Theory for Volterra Integrodifferential Systems", Journal of Differential Equations, Vol. 8, 1971, pp. 457-474.

[6] S. I. Grossman and R. K. Miller, "Nonlinear Volterra Integrodifferential Systems with L^1-kernels", Journal of Differential Equations, Vol. 13, 1973, pp. 551-566.

[7] G. S. Jordan and R. L. Wheeler, "Asymptotic Behavior of Unbounded Solutions of Linear Volterra Integral Equations," Journal of Mathematical Analysis and Applications (to appear).

[8] E. L. Lasley and A. N. Michel, "Input-Output Stability of Large-Scale Systems", Proceedings of the Eighth Asilomar Conference on Circuits, Systems, and Computers, Pacific Grove, California, December 1974, pp. 472-482.

[9] E. L. Lasley and A. N. Michel, "Input-Output Stability of Interconnected Systems", Proceedings of the IEEE International Symposium on Circuits and Systems, Boston, Mass., April 1975, pp. 131-134.

[10] E. L. Lasley and A. N. Michel, "L_∞- and l_∞-stability of Interconnected Systems" (to appear).

[11] R. K. Miller, Nonlinear Volterra Integral Equations, W. A. Benjamin, Menlo Park, California, 1971.

[12] R. K. Miller, "Asymptotic Stability Properties of Linear Volterra Integrodifferential Equations", Journal of Differential Equations, Vol. 10, 1971, pp. 485-506.

[13] R. K. Miller, "Structure of Solutions of Unstable Linear Volterra

Integrodifferential Equations", Journal of Differential Equations, Vol. 15, 1974, pp. 129-157.

[14] R. K. Miller and J. A. Nohel, "A stable Manifold Theorem for a System of Volterra Integrodifferential Equations," SIAM Journal of Mathematical Analysis, Vol. 6, 1975, pp. 506-522.

[15] A. Ostrowski, "Determinanten mit überwiegender Hauptidiagonale und die absolute Konvergenz von linearen Iterationprozessen", Comentarii Math. Helvitici, Vol. 30, 1956, pp. 175-210.

[16] H. Plaza and W. H. Kohler, "Coupled-reactor Kinetics Equations", Nuclear Science and Engineering, Vol. 22, 1966.

[17] D. W. Porter and A. N. Michel, "Stability of Multiple-Loop Nonlinear Time-Varying Systems", Report No. ISU-ERI-Ames-73167, Iowa State University, 1973.

[18] D. W. Porter and A. N. Michel, "Input-Output Stability of Time-Varying Nonlinear Multiloop Feedback Systems", IEEE Transactions on Automatic Control, Vol. 19, 1974, pp. 422-427.

[19] W. Rudin, Fourier Analysis on Groups, Interscience Publishers, N. Y., 1962.

[20] I. W. Sandberg, "On the L_2-boundedness of Solutions of Nonlinear Functional Equations", Bell System Tech. J., Vol. 43, part 2, July 1964, pp. 1581-1599.

[21] I. W. Sandberg, "Some Results on the Theory of Physical Systems Governed by Nonlinear Functional Equations", Bell Syst. Tech. J., Vol. 44, 1965, pp. 871-898.

[22] M. Vidyasagar, "L_2-stability and L_2-instability of Interconnected Feedback Systems", Proc. IEEE (to appear).

[23] J. C. Willems, The Analysis of Feedback Systems, MIT Press, Cambridge, Mass., 1971.

[24] G. Zames, "On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems - Part I: Conditions Derived Using the Concepts of Loop Gain, Conicity and Positivity", IEEE Trans. on Automatic Control, Vol. 11, 1966, pp. 228-238.

[25] G. Zames, "On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems - Part II: Conditions Involving Circles in the Frequency Plane and Sector Non-linearities", IEEE Transactions on Automatic Control, Vol. 11, 1966, pp. 465-476.

STABILITY OF MULTIVARIABLE FEEDBACK SYSTEMS CONTAINING ELEMENTS WHICH ARE OPEN-LOOP UNSTABLE

R. K. MILLER*
Department of Mathematics

Iowa State University
Ames, Iowa 50011

and

A. N. MICHEL**
Electrical Engineering Department and
Engineering Research Institute
Iowa State University
Ames, Iowa 50011

ABSTRACT

New results for the L_p-stability of feedback systems are established. Systems are considered which contain elements that may be open-loop unstable. These results are often simpler to apply than earlier results for such systems. A new Wiener-Lévy type theorem is proved. The results are applied to a class of "large scale" dynamical system (also called composite systems, interconnected systems or multi-loop systems) and to some specific examples.

I. INTRODUCTION

New results for the L_p-stability (or input-output stability on L_p) of feedback systems are established. Systems are considered which contain elements that may be open-loop unstable. The present results are motivated by recent work of Callier and Desoer [1], [2] and [4] and by Grossman and Miller [9, section V]. These results are in a form which can easily be applied to multiple input-multiple output systems (MIMO systems) with interconnections (also called interconnected systems, composite systems, multiloop systems and the like) when all elements of the interconnecting structure (but not necessarily of the subsystems) are stable. This will be illustrated below.

The results of Callier and Desoer concern systems with elements whose Laplace transforms may have poles of arbitrary order at a finite number of points s_k in the right half complex plane $\mathrm{Re}\, s \geq 0$. Their analysis depends on the theory of coprime decompositions of polynomial matrices. Their results are quite interesting but the algebra can be rather complicated for computation in high dimensional systems. Grossman and Miller proved some theorems of Wiener-Lévy type for integrodifferential equations. Their results can be viewed as an analysis of systems with unstable elements but the only pole allowed is $s = 0$. Their results are complete for a simple pole at $s = 0$. Only partial results were obtained for higher order poles at $s = 0$.

In the present paper the results in [9] are extended to necessary and sufficient conditions for stability when there is a pole at $s = 0$ of arbitrary order. In addition it is shown that for the purposes of stability analysis the type of systems studied in [1], [2] with poles at a finite set of points $\{s_1, s_2, ---, s_M\}$ can be reduced to systems with poles at $s = 0$. These results provide a completely different alternate way to analyze stability of systems which contain elements that are unstable. This alternate procedure for checking stability has the advantage that only straightforward

---

* This author was supported by the National Science Foundation under Grant No. MPS 75-07622.

** This author was supported by the National Science Foundation under Grant ENG 75-14093 and by the Engineering Research Institute, Iowa State University.

and rather elementary algebra is needed. Also products $(s - s_k)^{-j} g_k^*(s)$ are allowed in hypotheses (P) below but such terms are not allowed in [1], [2]. These products can often be employed to good advantage to reduce the dimension of phase space and thereby reduce computational complexity. Finally when $p = 2$, the L_p-stability conditions can have the practical advantage of being in such a form such that it is possible to use frequency domain techniques to verify the conditions.

For a summary of well known input-output stability results, originally established by Sandberg and by Zames, refer to [5] and [18]. See also the related work of Desoer and Wu [6], [7] and Wu and Desoer [19]. For input-output stability results of interconnected systems see Porter and Michel [15], [16] and Lasley and Michel [10], [11], [12]. For background information on Volterra integral equations see [13].

II. NOTATION AND PRELIMINARIES

Let $A = [a_{ij}]$ be an $m \times n$ matrix and A^T its transpose. $A = [a_{ij}] \geq B = [b_{ij}]$ means $a_{ij} \geq b_{ij}$ for all i and all j. Let E^m be real or complex n-space with elements $x = [x_1, x_2, ---, x_m]^T$ and norm $|x|$. $|A|$ is the induced norm on any square matrix A. $\|x\|$ will mean the vector $\|x\| = [|x_1|, |x_2|, ---, |x_m|]^T$. I is the identity matrix. For $1 \leq p < \infty$, L_p will be the usual Lebesgue space of functions $f: R^+ \to E^m$ with

$$|||f|||_p = [\|f_1\|_p, \|f_2\|_p, ---, \|f_m\|_p]^T$$

and

$$\|f_j\|_p = \{\int_0^\infty |f_j(t)|^p dt\}^{1/p} \quad (1 \leq p < \infty,\ p = \infty \text{ is similar})$$

for each component f_j of f. Thus $\|f\|_p \equiv |(|||f|||_p)|$ is the usual L_p-norm of a vector valued function f.

As is usual in systems theory L_{pe} is the extended L_p-space of functions $f: R^+ \to E^m$ whose truncations f_T belong to L_p for all $T > 0$. Let $C(R^+)$ be the space of all continuous functions $f: R^+ \to E^n$.

Let η be the set of all memoryless nonlinear functions $\varphi: R^+ \times E^m \to E^m$ such that $\varphi(t,x)$ is measurable in $t \in R^+$ for x fixed, continuous in $x \in E^m$ for each fixed $t \geq 0$, and $\varphi(t,0) = 0$ a.e. on $0 \leq t < \infty$. A function $\varphi \in \eta$ is called interior conic (F,R) for some matrix F and some nonnegative matrix R if $\|\varphi(t,x) - Fx\| \leq R\|x\|$ for all $x \in E^m$ and almost all $t \geq 0$. For $\varphi \in \eta$, φ is called incrementally interior conic (F,R) if

$$\|\varphi(t,x) - \varphi(t,y) - F(x - y)\| \leq R\|x - y\|$$

for all $x, y \in E^m$ and for almost all $t \geq 0$.

$\mathcal{L}_e$ will denote the set of all linear operators H on L_{pe} defined by convolution with kernel $h \in L_{1e}$, that is $H \in \mathcal{L}_e$ if and only if there is a matrix function $h \in L_{1e}$ such that for any $f \in L_{pe}$,

$$H f(t) = (h * f)(t) = \int_0^t h(t - s)f(s)ds$$

almost everywhere in $t \geq 0$. $\mathcal{L}$ is that subset of $\mathcal{L}_e$ for which the kernels h are in L_1.

Given a function $f: R^+ \to E^n$, $f^*(s)$ will denote its Laplace transform where $s = \sigma + j\tau$ is a complex variable and $j = \sqrt{-1}$.

A function $B: L_{pe} \to L_{pe}$ is called *stable* (or *open-loop stable*) on L_p if there is a constant $K \geq 0$ such that

$$\|(Bf)_T\|_p \leq K \|f_T\|_p$$

for all $T > 0$ and all $f \in L_{pe}$. The *gain* of B, denoted by $g_p(B)$, is the greatest lower bound of all such numbers K. For a matrix $B = [b_{ij}]$ define $g_p(B) = [g_p(b_{ij})]$. It is well known that for $H \in \mathcal{L}$ with kernel h, $g_p(H) \leq \|h\|_1 < \infty$. For $p = \infty$ the converse is also true [3, theorem 3]: if $H \in \mathcal{L}_e$, then H is (open-loop) stable on L_∞ if and only if $H \in \mathcal{L}$.

Consider the feedback system

$$\text{(S)} \quad \begin{aligned} e_1(t) &= u_1(t) - \varphi(t, e_2(t)) \\ e_2(t) &= u_2(t) + He_1(t) \end{aligned}$$

where u_1 and $u_2 \in L_{pe}$, $\varphi \in \eta$ and $H \in \mathcal{L}_e$ with kernel h. (S) is called *well posed* on L_{pe} if for each pair u_1, u_2 there is a unique solutions pair $e_1, e_2 \in L_{pe}$ and if the solutions depend continuously on u_1 and u_2. If φ is incrementally interior conic (F,R), then on L_{pe} for $1 \leq p < \infty$ the system (S) is well posed (see e.g., Miller [13, Chapter III]). (S) is called *L_p-stable* for some $p (1 \leq p \leq \infty)$ if for all pairs u_1 and u_2 in L_p all solutions e_1 and e_2 of (S) are in L_p.

III. MAIN RESULTS

The following theorem contains and generalizes the results in section V of [9].

Theorem 1. *Let* A_j *be constant matrices for* $j = 0, 1, ---, n$, $B_j \in L_1$ *be integrable matrix valued functions for* $j = 0, 1, ---, n+1$, *define*

$$a(t) = \sum_{j=0}^{n} (A_j t^{n-j}/(n-j)! + \int_0^t \int_0^{t_1} \int_0^{t_2} --- \int_0^{t_j} B_{n-j}(t_{j+1}) dt_{j+1} --- dt_1)$$

and define $L(t) = a(t) + B_{n+1}(t)$. Assume $A_0 + B_0(t) \not\equiv 0$. *Let* $r_0(t)$ *be the integral resolvent of* $L(t)$, *that is* r_0 *is the unique solution of*

$$\text{(1)} \quad r_0(t) = -L(t) + \int_0^t L(t-s) r_0(s) ds$$

and for $j = 0, 1, ---, n+1$ *let*

$$\text{(2)} \quad r_{n+1-j}(t) = \frac{d^j}{dt^j} \{t^n/n! - \int_0^t \int_0^{t_1} --- \int_0^{t_{n-1}} r_0(t_n) dt_n --- dt_1\}$$

or equivalently in terms of Laplace transforms

$$r_0^*(s) = -(I - L^*(s))^{-1} L^*(s), \quad r_j^*(s) = s^{-j}(I - L^*(s))^{-1}.$$

Then all of the following are equivalent.

(a) $\det s^{n+1}(I - L^*(s)) \neq 0$ *when* $\mathrm{Re}\, s \geq 0$.

(b) $r_j \varepsilon L_1$ for $0 \leq j \leq n+1$.

(c) $r_0 \varepsilon L_1$.

Remark 1. If $a(t) \equiv 0$ so that $n = -1$ and $L \equiv B_0 \varepsilon L_1$, then this theorem reduces to the following result of Paley and Wiener [17]: $r_0 \varepsilon L_1$ if and only if $\det(I - B_0^*(s)) \neq 0$ when $\mathrm{Re}\, s \geq 0$.

Remark 2. If $B_{n+1}(t) \equiv 0$, then the result was proved in [9].

Remark 3. For the general $L(t)$ it was shown in [9] that (a) implies (c).

Theorem 1 will be a main tool used to derive the remaining results. Consider an operator $H \varepsilon \mathcal{L}_e$ with kernel h whose Laplace transform has the form

$$\text{(P)}\quad h^*(s) = \sum_{k=1}^{N} \sum_{\alpha=0}^{m_k-1} M_{k\alpha}(s - s_k)^{-\alpha-1} + \sum_{k=N+1}^{M} \sum_{\alpha=0}^{m_k-1} M_{k\alpha}(s - s_k)^{-\alpha-1} g_k^*(s) + g_{M+1}^*(s)$$

Here $g_k \varepsilon L_1$ are matrix valued functions for $N+1 \leq k \leq M+1$, the $M_{k\alpha}$ are matrices and some or all $M_{k\alpha}$ may be zero. (The hypotheses of [1], [2] require that $M_{k\alpha} = 0$ for $N+1 \leq k \leq M+1$.) Let $p(s)$ be the least common divisor of the $(s - s_k)^{m_k}$, that is $p(s)$ is the polynomial of minimal degree and with leading coefficient equal to one such that $p_k(s) = (s - s_k)^{-m_k} p(s)$ is a polynomial for all $k = 1, 2, ---, M$.

Theorem 2. *Consider system* (S) *when* $\varphi \varepsilon \eta$ *is interior conic* (F,R) and $H \varepsilon \mathcal{L}_e$ *satisfies the pole condition* (P) *and the determinant condition*

$$\text{(D)}\quad \det(p(s)I + p(s)h^*(s)F) \neq 0 \text{ when } \mathrm{Re}\, s \geq 0.$$

Then $H_1 \equiv (I + HF)^{-1}H$ *exists and is in* $\mathcal{L}$. *Moreover if* $1 \leq p \leq \infty$ *and if*

$$\text{(3)}\quad m(p) = |g_p(H_1)|\ |R| < 1,$$

then (S) *is* L_p-*stable*.

Condition (D) can often be checked graphically by plotting the graph of the frequency function F defined by

$$F(j\omega) \equiv \det(p(j\omega)I + p(j\omega)h^*(j\omega)F)$$

in the complex plane as ω varies over $-\infty < \omega < \infty$. In the special case $p = 2$ the elements of the matrix $g_p(H_1)$ can often be obtained or estimated using frequency domain plots.

Corollary 1. *If* $\varphi \varepsilon \eta$ *is interior conic* (F, rI) *for some number* $r \geq 0$ *and if* H *satisfies* (P) *and* (D), *then* $H_1 \varepsilon \mathcal{L}$. *Also* (S) *will be* L_p-*stable if* $|g_p(H_1)| r < 1$.

The following result is more convenient to apply to composite systems.

Theorem 3. *Assume* $\varphi \varepsilon \eta$ *is interior conic* (F,R), *assume* $H \varepsilon \mathcal{L}_e$ *satisfies* (P) *and* (D), *define* $H_1 = (I + HF)^{-1}H$ *and* $M_p = I - g_p(H_1)R$. *Then* $H_1 \varepsilon \mathcal{L}$. *Moreover* *if* $1 \le p \le \infty$ *and if the successive principle minors of* M_p *are all positive, then* (S) *is* L_p*-stable*.

Of special interest in applications are MIMO systems of the form

$$(S') \qquad e_{1k} = u_{1k} - \varphi_k(t, e_{2k}) - \sum_{i=1}^{m} F_{ki}(t, e_{2i})$$

$$e_{2k} = u_{2k} + H_k e_{1k}$$

where $1 \le k \le m$, e_{1k} and e_{2k} are n_k-vector valued and $n = \sum_{i=1}^{m} n_k$. This system can be viewed as the interconnection of m *isolated subsystems* of the form

$$(S_k) \qquad e_{1k} = u_{1k} - \varphi_k(t, e_{2k})$$

$$e_{2k} = u_{2k} + H_k e_{1k}$$

for $k = 1,2,---,m$ with *interconnecting structure* specified by the functions F_{ki}.

Theorem 4. Let $\varphi_k \in \eta$ *be interior conic* $(F_k, r_k I)$, *let* $F_{ki} \in \eta$ *be interior conic* $(0, S_{ki} I)$ *and let* $H_k \in \mathcal{L}_e$ *satisfy conditions of the form* (P) *and* (D) *for each* $k = 1,2,---,m$. *Then* $K_k \equiv (I + H_k F_k)^{-1} H_k \in \mathcal{L}$ *for* $k = 1,2,---,m$. *Moreover if one defines* M_p *as the* $m \times m$ *matrix whose elements are*

$$m_{ij} = \delta_{ij}(1 - |g_p(K_i)| r_i) - |g_p(K_i)| s_{ij}$$

where δ_{ij} *is the Kronecker delta, if* $1 \le p \le \infty$ *and if the successive principle minors of* M_p *are all positive, then* (S′) *is* L_p*-stable*.

Finally note that for linear system (S) condition (D) is not only sufficient but also necessary for stability. Indeed the following result is immediate from Theorem 1 and 2.

Corollary 2. Let $\varphi(t,x) \equiv Fx$ *and let* $H \in \mathcal{L}_e$ *satisfy* (P). *Then* $H_2 = -(I + HF)^{-1} HF \in \mathcal{L}$ *if and only if* (D) *is true. If* (D) *is violated at a point* $s_1 = \sigma + j\tau$ *then* (S) *is* L_∞*-unstable when* $\sigma = 0$ *and* (S) *is* L_p*-unstable for all* $p \in [1, \infty]$ *when* $\sigma > 0$. *If* $H_2 \in \mathcal{L}$, *then* $H_1 = (I + HF)^{-1} H$ *must be in* $\mathcal{L}$.

Note that in Corollary 2 the kernel of the operator H_2 is the resolvent r_0 of Theorem 1. Also note that det $F = 0$ is allowed. That is the point of the last sentence in Corollary 2.

IV. EXAMPLES

Linear Volterra integrodifferential equations of order $n \ge 1$ provide examples of systems (S) with poles at the origin. This was already pointed out in [9]. These equations also are nice examples for illustrating the convenience of allowing product terms $(s - s_k)^{-j} g_k^*(s)$ in (P).

Callier and Desoer [1] studied linear systems $e_1 = He_2$, $e_2 = u_2 - Fu_1$, where F is a constant matrix. In one of their examples $F = I$, Re $s_1 > 0$, e_1 and e_2 are 2-vectors, and

$$h^*(s) = \begin{bmatrix} \frac{s - s_1}{s+1} & \frac{e^{-s}}{2} \\ \frac{e^{-s}}{s} & \frac{s+1}{s - s_1} \end{bmatrix} - I.$$

For this special case the polynomial $p(s)$ can be chosen as $p(s) = s - s_1$ or as $p(s) = (s - s_1)(s+1)$. Use the latter choice here so that (D) reduces to

$$\det p(s)(I + h^*(s)) = (s+1)(s - s_1)(1 - e^{-2s}/10) \neq 0 \text{ when } \operatorname{Re} s \geq 0?$$

Since this condition is violated at $s = s_1$, then by Corollary 2 the system is L_p-unstable. This simple, natural analysis avoids the computation of co-prime factors.

Some systems with delays can be analyzed with the results presented here. Let $t_0 = 0$, $t_{n+1} > t_n$ for $n = 0,1,2,---$, $\sum_{n=0}^{\infty} |k_n| < \infty$, $k \varepsilon L_1$ and define

$$\begin{aligned} K\varphi(t) &= \Big(\sum_{n=0}^{\infty} k_n \delta(t - t_n) + k\Big) * \varphi(t) \\ &= \sum_{n=0}^{\infty} k_n \varphi(t - t_n) + \int_0^t k(s)\varphi(t - s)\,ds \end{aligned}$$

where $\delta(t)$ is the delta function and for convenience $\varphi(t)$ is defined to be zero for t negative. Consider the equation

$$x'(t) + ax(t) = e^{bt} * x(t) + Kx(t) + f(t), \quad x(0) = x_0.$$

Then

$$\begin{aligned} x^*(s) = {} & (x_0 + f^*(s))/(s+a) + \{(s-b)^{-1}(s+a)^{-1} \\ & + \sum_{n=0}^{\infty} k_n e^{-st_n}/(s+a) + k^*(s)/(s+a)\}x^*(s) \end{aligned}$$

so that (P) is true. Also (D) reduces to

$$(s-b)(s+a) - \Big(\sum_{n=0}^{\infty} k_n e^{-st_n} + k^*(s)\Big)(s-b) - 1 \neq 0 \text{ when } \operatorname{Re} s \geq 0.$$

V. PROOFS

The theorems stated in section 3 will now be proved.

Proof of Theorem 1: Assume that (a) is true. In remark 3 above it was noted that (a) implies that $r_0 \varepsilon L_1$. So it is only necessary to show that $r_j \varepsilon L_1$ for $1 \leq j \leq n+1$ in order to see that (a) implies (b). Let b_{n+1} be a C^{∞}-smooth function with compact support in $[0,\infty)$ such that $\|b_{n+1} - B_{n+1}\|_1 < \mathcal{E}$. Choose $\mathcal{E}$ so small that

$$(4) \quad \det s^{n+1}(I + a^*(s) + b^*_{n+1}(s)) \neq 0 \text{ when } \operatorname{Re} s \geq 0.$$

This is possible since (a) is true. Let $L_1 = a + b_{n+1}$, let $\Delta B = B_{n+1} - b_{n+1}$, and let ρ be the integral resolvent of L_1, i.e., $\rho^*(s) = -(I - L_1^*(s))^{-1} L_1^*(s)$. Fix any given bounded function $\varphi \varepsilon C(R^+)$, define $f = 1 * \varphi$ and let x be the solution of the integral equation $x = f + L * x$. Then equivalently x solves

$$x = f + L_1 * x + \Delta B * x \quad \text{or} \quad (\delta - L_1) * x = f + \Delta B * x.$$

By the Paley-Wiener Theorem (remark 1) and (4) it follows that $\rho \varepsilon L_1$. Since $(\delta - \rho) * (\delta - L_1) = \delta$ and $f = 1 * \varphi$, then

$$x = (\delta - \rho) * f + (\delta - \rho) * \Delta B * x$$

or

$$(5) \quad x = \{(\delta - \rho) * 1\} * \varphi + (\delta - \rho) * \Delta B * x.$$

By remark 2 above it follows that $(\delta - \rho) * 1 \varepsilon L_1$. Since $\{(\delta - \rho) * \Delta B\}^*(s) = \{I - \rho^*(s)\}\Delta B^*(s)$, then

$$\begin{aligned} \det(I - \{(\delta - \rho) * \Delta B\}^*(s)) &= \det\{(I - L_1^*(s))^{-1}(I - L_1^*(s) - \Delta B^*(s))\} \\ &= \det(I - L_1^*(s))^{-1}\det(I - L^*(s)). \end{aligned}$$

The determinant is not zero when $\operatorname{Re}s \geq 0$ by (a) and (4). The Paley-Wiener Theorem implies that the solution of (5) can be written as

$$x = (\delta - R) * \{(\delta - \rho) * 1 * \varphi\} \equiv R_1 * \varphi,$$

where $R \varepsilon L_1$. If one defines $R_1 \equiv (\delta - R) * ((\delta - \rho) * 1)$, then clearly $R_1 \varepsilon L_1$. On the other hand the solution of

$$x = f + L * x, \quad f = 1 * \varphi$$

is

$$x = f - r_0 * f \equiv \{(\delta - r_0) * 1\} * \varphi.$$

Thus $(\delta - r_0) * 1$ must be equal to R_1. Since $R_1 \varepsilon L_1$ and $r_{n+1} = (\delta - r_0) * 1$, then $r_{n+1} \varepsilon L_1$.

A similar argument can be used for $r_n, r_{n-1}, ---, r_1$ successively. Let $f = \varphi * 1 * 1$ and let x solve

$$x = f + L * x = f + L_1 * x + \Delta B * x.$$

Then $x = (\delta - \rho) * f - (\delta - \rho) * \Delta B * x$. With $R \varepsilon L_1$ one again has

$$x = \{(\delta - R) * (\delta - \rho) * 1 * 1\} * \varphi \equiv R_2 * \varphi$$

and $R_2 \varepsilon L_1$. On the other hand x can be written as

$$x = (\delta - r_0) * f = \{(\delta - r_0) * 1 * 1\} * \varphi,$$

so $r_n = (\delta - r_0) * 1 * 1 = R_2 \varepsilon L_1$. Continue by induction to see that (a) implies (b).

Clearly (b) implies (c). To complete the proof assume (a) is false, that is assume there is a point s_0 with $\operatorname{Re}s_0 \geq 0$ and $\det s_0^{n+1}(I - L^*(s_0)) \neq 0$. This means that

$$r_0^*(s) = -\{s^{n+1}(I - L^*(s))\}^{-1}s^{n+1}L^*(s)$$

has a singularity at $s = s_0$. Thus r_0 cannot be in L_1. This argument shows that (c) implies (a).

It will now be shown that the general case (P) can be reduced to the special case of poles only at $s = 0$. The following lemma will be needed.

Lemma. *Assume* $\varphi(t,x) = Fx$ *is linear*, $H \varepsilon \mathcal{L}_e$ *satisfies* (P) *and* (D) *and* $h_1 = (\delta + hF)^{-1} * h$, $h_2 = \delta - (\delta + hF)^{-1}$. *Then* h_1 *and* $h_2 \varepsilon L_1$

Proof. Define $D(s) = h^*(s)p(s)$ so that

$$h_1^*(s) = (p(s)I + D(s)F)^{-1}D(s).$$

Since $D(s)$ is a polynomial of degree $< n + 1 =$ the degree of $p(s)$, then $h_1^*(s)$ is a finite linear combination of the functions r_j,

$$r_j^*(s) = (p(s)I + D(s)F)^{-1}s^{n+1-j}.$$

By Theorem 1 all of the functions $r_j \varepsilon L_1$. Thus $h_1 \varepsilon L_1$. Since $h_2 = h_1 F$, then $h_2 \varepsilon L_1$ too.

Proof of Theorem 2. Let D, h_1 and h_2 be as defined above. From (S) it follows that

$$\begin{aligned} e_2 &= (u_2 + Hu_1) - H\varphi(t,e_2) \\ &= (u_2 + Hu_1) - HFe_2 + H\{Fe_2 - \varphi(t,e_2)\} \end{aligned}$$

or

$$\begin{aligned} e_2 &= (I + HF)^{-1} u_2 + (I + HF)^{-1} H\{u_1 + Fe_2 - \varphi(t,e_2)\} \\ &= \{(\delta - h_2) * u_2 + h_1 * u_1\} - h_1 * (\varphi(t,e_2) - Fe_2). \end{aligned}$$

For any $T > 0$, estimate

$$|||e_{2T}|||_p \le g(\delta - h_2)\ |||u_2|||_p + g(h_1)\ |||u_1|||_p + g(h_1)R\ |||e_{2T}|||_p. \tag{6}$$

Take vector norms to get

$$\|e_{2T}\|_p \le (1 + \|h_2\|_1)\|u_2\|_p + \|h_1\|_1 \|u_1\|_p + |g(H_1)|\ |R|\ \|e_{2T}\|_p$$

or

$$\|e_{2T}\|_p \le (1 - m(p))^{-1}\{(1 + \|h_2\|_1)\|u_2\|_p + \|h_1\|_1 \|u_1\|_p\}$$

for all $T > 0$. Thus $e_2 \varepsilon L_p$. Since $e_1 = u_1 - \varphi(t,e_2)$, then

$$\begin{aligned} \|e_1\|_p &\le \|u_1\|_p + \|Fe_2 - \varphi(t,e_2)\|_p + \|-Fe_2\|_p \\ &\le \|u_1\|_p + |R|\ \|e_2\|_p + |F|\ \|e_2\|_p < \infty, \end{aligned}$$

and $e_1 \varepsilon L_p$. This proves Theorem 2.

Proof of Theorem 3. The proof here is the same as the proof of Theorem 2 down to line (6). At that point note that the hypotheses imply that $I - g_p(H_1)R$ has an inverse ρ which is nonnegative, see [8] or [14]. Thus for all $T > 0$

$$|||e_{2T}|||_p \le \rho\{g((I - HF)^{-1})\,|||u_2|||_p + g(H_1)\,|||u_1|||_p\} < \infty,$$

i.e., $e_2 \varepsilon L_p$. As before e_1 must also be in L_p when e_2 is.

Proof of Theorem 4. The proof is essentially the same as the proof of Theorem 3. The vectors $e_1 = [e_{11}, e_{12}, ---, e_{1n}]^T$ and $e_2 = [e_{21}, e_{22}, ---, e_{2n}]^T$ are grouped by components e_{jk} rather than by scaler components. After this regrouping the proof is the same.

VI. CONCLUDING REMARKS

At this point some additional remarks are in order.

1. The use of the Theorem 1 to analyze systems which contain unstable elements leads to a simpler analysis of such systems. In applications it is hoped that computational complexity will also be reduced.

2. Some of the results presented offer the advantage of allowing stability analysis of complex systems in terms of lower order subsystems and in terms of system interconnecting structure.

3. In certain special cases it is possible to check stability conditions for a problem by using frequency domain techniques (with all of the practical advantages which such techniques offer).

REFERENCES

1. F. M. Callier and C. A. Desoer, "Necessary and Sufficient Conditions for Stability for n-Input, n-Output Convolution Feedback Systems with a Finite Number of Unstable Poles", IEEE Trans. AC - 18, 1973, pp. 295-298.

2. F. M. Callier and C. A. Desoer, "L^p-stability ($1 \leq p \leq \infty$) of Multivariable Nonlinear Time-Varying Feedback Systems that are Open-Loop Unstable", Int. J. Control, Vol. 19, 1974, pp. 65-72.

3. C. Corduneanu, "Problémes globaux dans la théorie des équations intégrales de Volterra", Annali di Mat. Pura. e. Appl., Vol. 67, 1965, pp. 349-363.

4. C. A. Desoer and F. M. Callier, "Convolution Feedback Systems", SIAM J. Control, Vol. 10, 1972, pp. 737-746.

5. C. A. Desoer and M. Vidyasagar, Feedback Systems: Input-Output Properties, Academic Press, N. Y., 1975.

6. C. A. Desoer and M. Y. Wu, "Stability of Linear Time-Invariant Systems", IEEE Trans. Circuit Theory, Vol. 15, 1968, pp. 245-250.

7. C. A. Desoer and M. Y. Wu, "Stability of Multiple-loop feedback linear time-invariant systems", J. Math. Anal. Appl., Vol. 23, 1968, pp. 121-130.

8. M. Fiedler and V. Ptak, "On Matrices with Non-positive off-diagonal Elements and Positive Principle Minors", Czech. Math. J., Vol. 12, 1962, pp. 382-400.

9. S. I. Grossman and R. K. Miller, "Nonlinear Volterra Integrodifferential Systems with L^1-Kernels", J. Diff. Equat., Vol. 13, 1973, pp. 551-566.

10. E. L. Lasley and A. N. Michel, "Input-Output Stability of Large-Scale Systems", Proceedings of the Eighth Asilomar Conference on Circuits, Systems and Computers, Pacific Grove, California, December 1974, pp. 472-482.

11. E. L. Lasley and A. N. Michel, "Input-Output Stability of Interconnected Systems", Proceedings of the IEEE International Symposium on Circuits and Systems, Boston, Mass., April 1975, pp. 131-134.

12. E. L. Lasley and A. N. Michel, "L_∞- and ℓ_∞-Stability of Interconnected Systems", (to appear).

13. R. K. Miller, Nonlinear Volterra Equations, W. A. Benjamin, Menlo Park, California, 1971.

14. A. Ostrowski, "Determinanten mit überwiegender Hauptdiagonale und die absolute Konvergenz von linearen Iterationprozessen", Comentarii Math. Helvitici, Vol. 30, 1956, pp. 175-210.

15. D. W. Porter and A. N. Michel, "Stability of Multiple-Loop Nonlinear Time-Varying Systems", Report No. ISU-ERI-Ames-73167, Iowa State University, 1973.

16. D. W. Porter and A. N. Michel, "Input-Output Stability of Time-Varying Nonlinear Multi-loop Feedback Systems", IEEE Transactions on Automatic Control, Vol. 19, 1974, pp. 422-427.

17. R.E.A.C. Paley and N. Wiener, Fourier Transforms in the Complex Domain, Amer. Math. Soc., Providence, R.I., 1934.

18. J. C. Willems, The Analysis of Feedback Systems, MIT Press, Cambridge, Mass., 1971.

19. M. Y. Wu and C. A. Desoer, "L^p-Stability ($1 \le p \le \infty$) of Nonlinear Time-Varying Feedback Systems", SIAM J. Control, Vol. 7, 1969, pp. 356-364.

ANALYSIS OF A CLASS OF NONLINEAR SYSTEMS WITH A MULTIPLICATIVE FEEDBACK

ROBERT A. CURTIS
Rome Air Development Center
Rome, N.Y.

TING N. LEE
The George Washington Univ.
Washington, D.C.

ABSTRACT

The analysis of a class of nonlinear systems consisting of a linear time-invariant system in the forward path and a nonlinear gain control element in the multiplicative feedback path is presented in this paper.

Starting with the system's transfer function, an iterative solution is obtained with the nonlinear characteristic satisfying some boundedness and smoothness conditions. The uniqueness and the periodicity of this solution are also discussed. In addition, a distortion criterion is presented to insure the closeness of using the first order approximation in the steady state case.

INTRODUCTION

Previous work by Oliver (2) and later work by Davis (1), formulate and analyse the AGC (Automatic Gain Control) system in terms of differential equations, where an input carrier frequency is considerably greater than the envelope frequency. Stability criteria, at best, is formulated in terms of linearized differential equations that are extremely difficult to analyse. This paper obtains generalized gain requirements that guarantee a unique and continuous transfer operator. Provided the same criteria are met, the steady state output will be the same period as an input periodic signal.

In addition to the gain criteria, a distortion criterion is introduced to insure that the overall AGC linear transfer characteristics are greater than the nonlinear transfer characteristics.

NOTATION AND PRELIMINARIES

The overall AGC system is formulated as illustrated in Fig. 1. The forward linear amplifier is formulated in terms of an impulse response g(t) or the operator notation G(.). The feedback path is given by the nonlinear function f(.) which will be assumed to satisfy the following magnitude relation,

$$0 < \beta_1 \leq f(.) \leq \beta_2 \tag{1}$$

where β_1 is some small positive constant indicative of the saturation of a gain control element. The constant β_2 shall be considered to be one with no loss of generality. The corresponding operator to f(.) will be denoted by F(.). The input is denoted by r(t) and the output is denoted by y(t). Instead of the usual additive feedback, the system illustrated in Fig. 1, uses multiplicative feedback denoted by π .

In order to make the operator F(.) more amenable to analysis, it will be split into an average D.C. term, denoted by s_o , plus an A.C. term denoted by F´(.). Without any loss of generality, the feedback operator can be rewritten as follows,

$$F(.) = s_o + F'(.) \tag{2a}$$

where it is assumed that,

$$F'(0) = 0 \tag{2b}$$

The operator T(.) will denote the overall transfer operation from input, r(t), to the output, y(t). The symbol, $\|.\|$, will indicate an L_2 norm, and the standard symbol, $|.|$, will be used for the L_∞ norm or magnitude.

The maximum input magnitude of r(t) will be denoted in this paper by r_1 , such that,

$$r_1 = \sup|r(t)| \tag{3}$$

Using Fig. 1, the following equations are easily obtained,

$$y = G(e) \tag{4a}$$

$$s = F(y) \tag{4b}$$

$$e = sr \tag{4c}$$

$$y = T(r) \tag{4d}$$

Simple substitution in the above equations yields the basis for the iterative solution as,

$$G(r\ F(T(r))) = T(r). \tag{5}$$

An inherent feature of the AGC system is that for a stable, linear time-invariant amplifier, the output is bounded for bounded input. To show this, write the system equation as,

$$y(t) = \int_0^t g(v)r(t - v)f(y(t-v))dv. \tag{6}$$

Taking the magnitude on each side yields,

$$|y(t)| \leqslant \int_0^t |g(v)||r(t - v)|f(y(t - v))dv. \tag{7}$$

Since the stable impulse response g(t) satisfies,

$$\int_0^\infty |g(t)|dt < \infty \tag{8}$$

it follows using relations 1 and 3, that the output y(t) is bounded by,

$$|y(t)| \leqslant r_1 \int_0^\infty |g(t)|dt \quad , \tag{9}$$

and hence the AGC system inherently has a bounded output for a bounded input.

ANALYSIS OF A CLASS OF NONLINEAR SYSTEMS WITH MULTIPLICATIVE FEEDBACK

The following theorem provides an iterative solution to the AGC system, given that the linear operator is bounded, and that certain gain and input constraints are met.

Theorem 1: Let the AGC system be given as in Fig. 1, with the operators as defined in the Notation and Preliminaries. Define a sequence $T_n(r)$, such that,

$$T_n(r) = s_o G(r) + G(rF'(T_{n-1}(r))) \quad \forall \ n=1,2,..... \tag{10a}$$

and,

$$T_o(r) = 0 \quad . \tag{10b}$$

If G(.) is a bounded linear operator and the following gain constraints are satisfied in L_2 space,

$$||G(x) - G(y)|| \leqslant \alpha ||x - y|| \qquad \alpha > 0 \tag{11a}$$

$$||F'(x) - F'(y)|| \leqslant \beta ||x - y|| \qquad \beta > 0 \tag{11b}$$

where x and y are any two points in the L_2 space, then if the following is satisfied,

$$\alpha \beta |r_1| < 1 \tag{12}$$

where r_1 is the supremum input value as in equation 3 then, the sequence $T_n(r)$ converges uniquely to the transfer operator T(r).

Proof: Consider the normed difference,

$$||T_n(r) - T_m(r)|| = ||T_n(r) - T_{n-1}(r) + T_{n-1}(r) -.... \\ .... + T_{m+1}(r) - T_m(r)||, \quad \forall \ n > m > 0 \tag{13}$$

Using the triangle inequality, we have,

$$||T_n(r) - T_m(r)|| \leqslant ||T_n(r) - T_{n-1}(r)|| + ||T_{n-1}(r) - T_{n-2}(r)|| \\ +......+ ||T_{m+1}(r) - T_m(r)|| \quad . \tag{14}$$

Considering two adjacent terms and the relations 11, results in,

$$\|T_m(r) - T_{m-1}(r)\| = \|G(rF'(T_{m-1}(r)))-G(rF'(T_{m-2}(r)))\|$$
$$\leqslant \alpha\beta\,|r_1|\;\|T_{m-1}(r) - T_{m-2}(r)\| \qquad (15)$$

Now using the above (m-1) times, it follows that,

$$\|T_m(r) - T_{m-1}(r)\| \leqslant (\alpha\beta\,|r_1|)^{m-1}\,\|T_1(r)\|\;. \qquad (16)$$

Substituting equation 16 for each term on the right side of equation 14 yields,

$$\|T_n(r) - T_m(r)\| \leqslant (\alpha\beta\,|r_1|)^m\;\frac{1 - (\alpha\beta\,|r_1|)^{n-m}}{1 - (\alpha\beta\,|r_1|)}\;\|T_1(r)\| \qquad (17)$$

Hence, there exists N < m < n such that,

$$\|T_n(r) - T_m(r)\| \leqslant \frac{(\alpha\beta\,|r_1|)^N}{1 - \alpha\beta\,|r_1|}\;\|T_1(r)\| \qquad (18)$$

As G(.) is bounded by assumption, it follows that T (r) is bounded from equation 10a, and hence equation 18 satisfys the Cauchy requirement, since,

$$\alpha\beta\,|r_1| < 1\;,$$

by hypothesis.

Next it is necessary to show that the limit of $T_n(r)$ belongs to the same normed metric space so that the space is complete and that T(r) is unique. Define the operator T(.), such that,

$$T(r) = \lim_{n\to\infty} T_n(r)\;. \qquad (19)$$

Now consider the norm,

$$\|s_oG(r) + G(rF'(T(r))) - T(r)\|$$
$$\leqslant \|s_oG(r) + G(rF'(T(r))) - (s_oG(r) + G(rF'(T_n(r))))\|$$
$$+ \|s_oG(r) + G(rF'(T_n(r))) - T(r)\|$$
$$\leqslant \|G(rF'(T(r))) - G(rF'(T_n(r)))\| + \|T_{n+1}(r) - T(r)\|$$
$$\leqslant \alpha\beta\,|r_1|\;\|T(r) - T_n(r)\| + \|T_{n+1}(r) - T(r)\| \qquad (20)$$

Now since the right hand side above can be made arbitrarily small, by relation 19, therefore the left hand side must be zero, implying,

$$T(r) = G(rF(T(r))). \qquad (21)$$

Hence T(r) is a solution of the system.

To show that the solution T(r) is unique, assume that another solution T'(r) exists, then,

$$\|T(r)-T'(r)\| \leqslant \|G(rF'(T(r)) - G(rF'(T'(r)))\|$$
$$\leqslant \alpha\beta\,|r_1|\;\|T(r) - T'(r)\| \qquad (22)$$

but the above equation can only be satisfied if T(r) equals T'(r), which implies that the solution T(r) is unique. This completes the proof.

To illustrate the iterative procedure, the following example will be considered.

Example 1: Consider an AGC system as in Fig. 1 with the linear time-invariant operator(in the frequency domain) given as,

$$k/(s + a) = G(s) \qquad . \qquad (23)$$

Assume that the input is a sine wave,

$$r(t) = r_1 \sin\omega t \qquad . \qquad (24)$$

Assume that the transfer characteristics of the operator F'(.) are given as F_{11}, ϕ_{11}, F_{12}, ϕ_{12}, etc. for the Fourier fundamental and second harmonic magnitude and phase angle due to a sine wave input. The initial estimate for s_o, will be assumed given, although a first order magnitude approximation to s_o, can be obtained by envelope considerations.

Solution: The steady state solution, y (t), will be obtained by the iterative equations 10a and 10b. The first iteration yields,

$$y_1(t) = T_1(r(t)) = w \sin(\omega t + E) \quad , \tag{25}$$

where w and E are given by,

$$w = (s_o k\, r_1)/\sqrt{\omega^2 + a^2} \tag{26a}$$

and,

$$E = -\tan^{-1}\omega/a \quad . \tag{26b}$$

The second iteration yields the distortion terms present in the output and is given by,

$$y_2(t) = w \sin(\omega t + E) + G(r_1 \sin\omega t\, F'(w \sin(\omega t + E))) \tag{27}$$

From computer tables, the Fourier magnitude, F_{1i}, and phase, ϕ_{1i}, transmission coefficients are obtained for a sine wave input to the nonlinear feedback circuit yielding,

$$\begin{aligned} F'(w \sin(\omega t + E)) = {} & F_{11}\, w \sin(\omega t + E + \phi_{11}) \\ & + F_{12}\, w \sin(2\omega t + E + \phi_{12}) \\ & \quad + F_{13}\, w \sin(3\omega t + E + \phi_{13}) \ldots . \end{aligned} \tag{28}$$

Equation 28 is substituted back into equation 27 and the resulting terms when operated on by G(.) yield the steady state solution,

$$\begin{aligned} y_2(t) = {} & w \sin(\omega t + E) + (r_1\, wF_{11}\, k/2a)\cos(E + \phi_{11}) \\ & + (r_1\, wF_{12}\, k/(2\sqrt{\omega^2 + a^2}))\cos(\omega t + 2E + \phi_{12}) \\ & - (r_1\, wF_{11}\, k/(2\sqrt{4\omega^2 + a^2}))\cos(2\omega t + E + \phi_{11} + \tan^{-1} 2\omega/a) \\ & + \ldots\ldots\ldots \end{aligned} \tag{29}$$

The next iteration would require the transfer coefficients for F'(.) due to several harmonic inputs. Notice that the main effect of the multiplicative feedback has been to introduce harmonic distortion terms as well as a phase shift to the linear term.

The next theorem considers the effect of a periodic signal on the operator and the output as a result of a periodic input.

Theorem 2: Let the transfer operator T(r) of the AGC system given in Fig. 1, satisfy equation 4 with G(.) and F(.) satisfying the previous boundedness criteria. If the hypothesis of Theorem 1 is satisfied, then a periodic input signal such that,

$$r(t) = r(t + t_p) \tag{30}$$

results in a periodic steady state output signal of the same period, t_p, such that,

$$T(r(t)) = T(r(t + t_p)) \ . \tag{31}$$

Proof: It follows from equation 10a with inputs r(t) and $r(t + t_p)$,

$$T(r(t)) = s_o G(r(t)) + G(r(t)F'(T(r(t))) \tag{32a}$$

$$T(r(t + t_p) = s_o G(r(t+t_p)) + G(r(t+t_p)F'(T(r(t+t_p))) \tag{32b}$$

As G(.) is a linear time invariant system then in the steady state,

$$G(r(t)) = G(r(t + t_p)) \tag{33}$$

Taking the difference norm of equations 32 and using equation 33, the following results,

$$\begin{aligned} \|T(r(t)) - T(r(t + t_p))\| \leqslant {} & \|G(r(t)F'(T(r(t))) \\ & - G(r(t + t_p)F'(T(r(t + t_p)))\| \ . \end{aligned} \tag{34}$$

Using conditions 11 and the fact that r(t) is periodic, equation 34 becomes,

$$\|T(r(t))-T(r(t+t_p))\| \leqslant \alpha\beta |r_1| \|T(r(t))-T(r(t+t_p))\| \tag{35}$$

This results in,

$$T(r(t)) = T(r(t + t_p)) \quad , \tag{36}$$

and therefore the output is periodic of period t_p. This completes the proof.

One of the requirements for an AGC amplifier is for the steady state output to be some constant multiple of the linear amplifier input without any noticeable distortion in the output. The effect of the multiplicative nonlinear feedback in an AGC system is to distort the steady state output. The steady state cost function, J, considered in this paper, minimizes the difference between the overall operator and an ideal linear operator, which is obtained by multiplying G(r(t)) by a constant $s_o + n$. The operators will be assumed to yield only real outputs in the following analysis. The cost equation is given by,

$$J= (1/t_p)\int_0^{t_p}(G(r(s_o + F'(T(r)))) - G(r(s_o + n)))^2 \, dt \tag{37}$$

Hence any deviation of y(t) from a linear transformation of the input, r(t), contributes to the cost, J. The following theorem derives the minimum cost for a given F′(.) as well as the minimum cost.

Theorem 3: For the AGC system as given in Fig. 1, let the distortion cost equation be given as in equation 37. Let the input, r(t), be periodic of period, t_p. Then for a given F′(.), the value of n that yields the minimum cost is given by,

$$n = \frac{\int_0^{t_p} G(r)G(rF'(T(r)))dt}{\int_0^{t_p} G(r)G(r)\, dt} \quad , \tag{38}$$

with the minimum cost given (trivially) by,

$$F'(.) = 0 \quad . \tag{39}$$

Proof: Differentiate the cost equation 37 with respect to n and set the result to zero to obtain,

$$0 = \int_0^{t_p} G(r)G(r(F'(.) - n))dt \quad . \tag{40}$$

Rearranging equation 40 and using the fact that n is a constant yields the result 38. The result 39 follows trivially by substituting 39 and 40 into the cost equation 37.

The fact that F′(.) equals zero for steady state minimum cost is hardly suprising. However, since F′(.) is not identically zero in most practical systems, a criterion must be established so that the linear transfer characteristics are larger than the nonlinear transfer characteristics in a steady state condition. The following relation is required,

$$\int_0^{t_p}(s_o G(r) + G(rF'(.)) - (n + s_o)G(r))^2 \, dt \leqslant (s_o + n)^2\int_0^{t_p} G(r)G(r)dt \quad , \tag{41}$$

where n is as given in Theorem 3. The following theorem provides a sufficient condition for the linear transfer part to be greater than the nonlinear transfer part.

Theorem 4: Let the AGC system be given as in Fig. 1, with the output of the AGC system given as,

$$T(r) = s_o G(r) + G(rF'(T(r))) \quad , \tag{42}$$

where s_o, $F'(.)$ and $G(.)$ are as given in the Notation and Preliminaries. Assume that the input, $r(t)$, is periodic of period t_p. Then the requirement for relation 41 to be satisfied, where n is as given in Theorem 3, is for the following constraint to be satisfied,

$$\int_0^{t_p} G(rF')G(rF')dt \leqslant (s_o^2/2) \int_0^{t_p} G(r)G(r)dt \tag{43a}$$

Or as $G(rF')$ may be difficult to find, the following is a sufficient condition,

$$\sup|F'(T(r))| \leqslant s_o/\sqrt{2} \quad . \tag{43b}$$

Proof: Substitute the value of n as derived in Theorem 3 into relation 41 to obtain, after some algebra,

$$\int_0^{t_p} G(r)G(r)dt \cdot \int_0^{t_p} G(rF')G(rF')dt \leqslant \left(s_o \int_0^{t_p} G(r)G(r)dt\right)^2$$
$$+ \; 2s_o \int_0^{t_p} G(r)G(r)dt \int_0^{t_p} G(rF')G(r)dt + 2\left(\int_0^{t_p} G(rF')G(r)dt\right)^2 \tag{44}$$

Recognizing that the right side of equation 44 is a parabola, relation 43a follows after minimizing the right side of 44.

In order to obtain a simpler condition the maximum value of $F'(.)$ is taken out of the integrand on the left side of relation 43a as,

$$\int_0^{t_p} G(rF')G(rF')dt \leqslant (\sup|F'(.)|)^2 \int_0^{t_p} G(r)G(r)dt \quad , \tag{45}$$

where it is assumed that $G(.)$ is a linear operator. In order to guarantee relation 43a the following is required,

$$(\sup|F'(.)|)^2 \int_0^{t_p} G(r)G(r)dt \leqslant (s_o^2/2) \int_0^{t_p} G(r)G(r)dt \quad , \tag{46}$$

and hence relation 43b follows. This completes the proof.

In order to illustrate the distortion criterion the following example is given.

Example 2: Consider the AGC system as in Fig. 1 with the linear time-invariant operator as given in Example 1. From the previous example $F'(.)$ is obtained assuming a sine wave input to the nonlinear feedback. Using equation 28 and assuming a worst case condition of phase angles results in,

$$\sup F'(.) = w \sum_{i=1}^{\infty} F_{1i} \tag{47}$$

Substituting into relation 43b and a little algebra results in,

$$r\,(k/\sqrt{\omega^2 + a^2}) \sum_{i=1}^{\infty} F_{1i} \leqslant 1/\sqrt{2} \tag{48}$$

which is slightly stricter than the gain constraint of Theorem 1.

CONCLUSION

In conclusion, the results of this paper provide new gain criteria to ensure a unique solution provided that an easily tested gain criterion is met. In addition, a periodicity requirement, guarantees that no subharmonic oscillations exist in the output solution. A distortion criterion is then introduced for measuring the nonlinear transfer characteristics compared to the linear transfer characteristics in the steady state output. The results presented in this paper provide fairly easy methods for testing and designing AGC systems.

SELECTED BIBLIOGRAPHY

1. Davis, R. C. "Mathematical Analysis of Automatic Gain Control Circuits." WESCON, (August 1961), 28/2.
2. Oliver, B. M. "Automatic Volume Control as a Feedback Problem." Proc. of IRE, 36(April 1948), 466-473.
3. Victor, W. K., and Brockman, M. H. "The Application of Linear Servo Theory to the Design of AGC Loops." Proc. of IRE, 48(February 1960), 234-238.
4. Zelenz, M. L. "On AGC Transient Response Behavior in Cascaded Amplifiers." Proc. of the IEEE, 58(July 1970), 1130-1131.

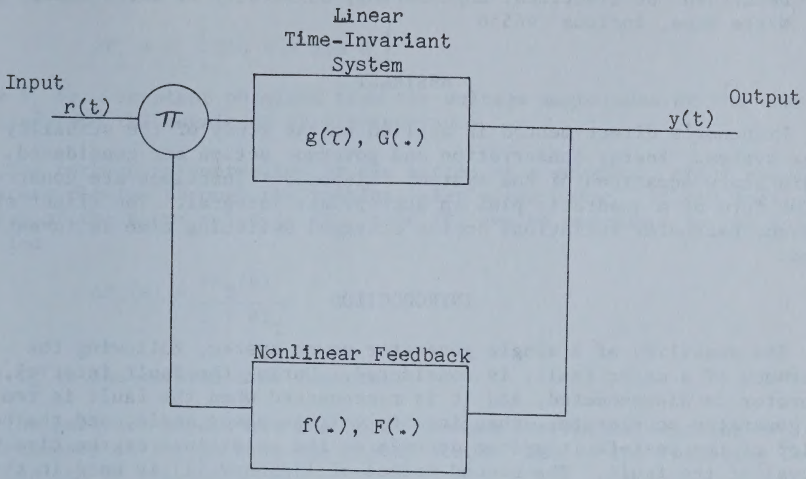

Linear
Time-Invariant
System
Input
r(t)
π
g(τ), G(.)
Output
y(t)
Nonlinear Feedback
f(.), F(.)

Fig. 1.--AGC System

STABILITY OF A POWER SYSTEM - THE EFFECT OF GOVERNOR CONTROL ACTION

L. JENKINS AND R. LIU
Department of Electrical Engineering, University of Notre Dame,
Notre Dame, Indiana 46556

ABSTRACT

Liapunov's direct mehtod is applied to the study of the stability of a power system. Energy conservation and governor action are considered, to obtain state equations of the system. Liapunov functions are constructed in the form of a quadratic plus an appropriate integral. The effect of governor parameter variations on the critical switching time is investigated.

INTRODUCTION

The stability of a single generator power system, following the occurence of a major fault, is considered. During the fault interval, the generator is disconnected, and it is reconnected when the fault is removed. The generator accelerates, changing its voltage phase angle, and the behavior of the post-fault system depends on the conditions at the time of removal of the fault. The second method of Liapunov [1] is used in the study.

In most previous studies [2,3,4] the effect of governor action has been assumed to be negligible. In this paper Liapunov functions which would take into account governor action are sought, so as to enable one to study the effect on the stability of variations in the governor feedback parameters.

MODEL OF THE POWER SYSTEM

The power system under consideration consists of a single synchronous generator connected through a transmission line to an infinite bus. The generator is driven by a steam turbine, and has a damper winding, the damping torque being assumed to be proportional to the deviation in the frequency from the steady state value.

Let

$$k_F \triangleq \frac{\pi F_o}{S_o} \tag{1}$$

where S_o is the energy stored in the generator shaft at the steady state frequency F_o. From energy considerations, the following differential equation is obtained [5]

$$\frac{d^2\delta}{dt^2} = k_F \left[\Delta P_T - \Delta P_G - D_a \frac{d\delta}{dt} \right] \tag{2}$$

where δ is the phase angle of the voltage phase of the generator with respect to the infinite bus (reference) voltage, ΔP_T and ΔP_G are the incremental turbine power flow and generator power flow, and D_a is the damping coefficient.

The voltage control feedback loop of the generator is assumed to have a sufficiently small time constant, such that variations in the voltage magnitude can be neglected. Frequency deviations are quite small and the transmission line highly inductive, hence the line can be represented by a constant reactance. Let δ_o be the steady state value of δ. Then ΔP_G is given by

$$\Delta P_G = k_L \left[\text{Sin } \delta - \text{Sin } \delta_o \right] \tag{3}$$

where k_L is a constant obtained from the voltage magnitudes at the ends of the line, and the magnitude of its reactance.

The turbine is controlled by the action of a governor, which regulates the steam flowing through the turbine inlet steam valve. If the incremental setting of the steam valve is ΔX_E, then ΔP_T can be expressed by the Laplace equation.

$$\Delta P_T(s) = \frac{\Delta X_E(s)}{1 + sT_T} \tag{4}$$

where T_T is the turbine time constant.

The governor inputs are the speed changes incremental setting ΔP_c and the frequency deviation from the steady state. Hence, the following Laplace equation can be written

$$\Delta X_E(s) = \frac{\Delta P_c(s) - \beta \Delta F(s)}{(1+sT_G)} \tag{5}$$

where B is a constant known as the regulation constant, and T_G is the time constant of the governor. ΔP_c is chosen from the point of view of maintaining a constant frequency in the system when small load pertubation take place. Hence integral control is often used, and ΔP_c is given by

$$\Delta P_c = -K_I \int_o^t \Delta F dt \tag{6}$$

where K_I is a positive gain constant. ΔF can be express in terms of δ.

$$\Delta F = \frac{1}{2\pi} \frac{d\delta}{dt} \tag{7}$$

Let

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \begin{matrix} = \\ = \\ = \\ = \end{matrix} \begin{bmatrix} \delta - \delta_o \\ \frac{d\delta}{dt} \\ \Delta X_E \\ \Delta P_T \end{bmatrix} \tag{8}$$

Using (6) and (7), then (2), (4) and (5) can be put in the state equation form

$$\dot{\underline{x}} = A\underline{x} + Bf(\underline{x}) \tag{9}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -D_a k_F & 0 & k_F \\ \frac{-K_I}{2\pi T_G} & \frac{-\beta}{2\pi T_G} & -\frac{1}{T_G} & 0 \\ 0 & 0 & \frac{1}{T_T} & -\frac{1}{T_T} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

$$f(\underline{x}) = k_F k_L \left[\text{Sin } (x_1 + \delta_o) - \text{Sin } \delta_o \right] \tag{11}$$

THE SECOND METHOD OF LIAPUNOV

The asympotic stability is studied via the second method of Liapunov [1]. LaSalle's theorem is applied.

Theorem

Let $V(\underline{x})$ be a Liapunov function. Let D_s be a region in which $V(\underline{x}) < \ell_o$ and $\dot{V}(\underline{x}) \leq 0$. Let R_D be the set of points in D_s for which $\dot{V}(\underline{x}) = 0$, and let M_o be the largest invariant set in R_D. Then every solution $\underline{x}(t)$ in D_s tends to M_o as time tends to infinity.

For our purpose, D_s is chosen such that M_o has only one point, the origin. Thus, if the initial condition, at the time of removal of the fault, lies within D_s, then the asympotic stability of the solution is assured.

Let critical switching time be the time elapsed from the instant at which the fault occurs, until the instant at which the faulted system trajectory leaves the region D_s. Therefore, the critical switching time determines the fault interval for which the asympotic behavior of the post fault solution is guaranteed. Since Liapunov's method provides a sufficient but not a necessary condition for stability, the critical switching time obtained depends on the choice of Liapunov function. The critical switching time can be found by simulation of the faulted system on a digital computer. A measure of the suitability of the Liapunov function chosen is the discrepancy between the critical switching time and the maximum switching time at which stable behavior is observed in the simulated swing curves.

GENERATOR WITHOUT FEEDBACK

This case has been considered in [2] and [4]. The results are summarized here for completeness, and to indicate the approach to be used when feedback is included.

In the case of without feedback, both ΔX_E and ΔP_T are identically zero. Hence (9) becomes

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 1 \\ 0 & -D_a k_F \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{12}$$

The Liapunov function is chosen, from energy considerations, as an integral plus a quadratic form

$$V(\underline{x}) = x_2^{\,2} + a \int_o^{x_1} [\mathrm{Sin}\ (\xi + \delta_o) - \mathrm{Sin}\ \delta_o] d\xi \tag{13}$$

where

$$a = 2k_F k_L \tag{14}$$

$\dot{V}(\underline{x})$ is found to be negative semidefinite for all positive values of D_a. In the application of LaSalle's theorem, the region D_s is chosen as the open set, including the origin, and bounded by the curve

$$V(\underline{x}) = V_B \tag{15}$$

where

$$V_B = V(\underline{x}_u) \tag{16}$$

$$\underline{x}_u = \begin{bmatrix} \pi - 2\delta_o \\ 0 \end{bmatrix} \tag{17}$$

The critical switching time is found to match closely the maximum switching time found by simulation.

GENERATOR WITH $K_I = 0$

In this case, the matrix A of equation (9) reduces to

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -D_a k_F & 0 & k_F \\ 0 & -\dfrac{\beta}{2\pi T_G} & -\dfrac{1}{T_G} & 0 \\ 0 & 0 & \dfrac{1}{T_T} & \dfrac{-1}{T_T} \end{bmatrix} \tag{18}$$

Let us first consider Liapunov function of the same form as (13).

$$V(\underline{x}) = a_1 \int_o^{x_1} [\mathrm{Sin}\ (\xi + \delta_o) - \mathrm{Sin}\ \delta_o]\ d\xi + a_2 x_2^{\,2} + a_3 x_3^{\,2} + a_4 x_4^{\,2} \tag{19}$$

The a_i are chosen to make $\dot{V}(\underline{x})$ negative semidefinite. It can be shown that if

$$a_1 = k_L \qquad a_2 = \frac{1}{2k_F}$$

$$a_3 = \frac{\pi T_G}{\beta} \qquad a_4 = \frac{T_T}{4D_a} \tag{20}$$

then $V(\underline{x})$ is a Liapunav function whenever

$$\frac{8\pi D_a}{\beta} > 1 \tag{21}$$

The boundary of the stable region D_s, is chosen as the constant V curve

$$V_B = V(\underline{x}) \tag{22}$$

where

$$\underline{x}_u = \begin{bmatrix} \pi - 2\delta_o \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{23}$$

It is found that, for typical values of β, the discrepancy between the critical switching time and the maximum switching time obtained by simulation is quite large. Also the critical switching time does not change over a large range of β. Hence we consider the Liapunov function

$$V(\underline{x}) = a_1x_1^2 + a_2x_2^2 + a_3x_3^2 + a_4x_4^2 + a_5x_1x_2 + a_6x_1x_3 + a_7x_1x_4$$
$$+ a_8 \int_o^{x_1} [\text{Sin } (\xi + \delta_o) - \text{Sin } \delta_o] d\xi \tag{24}$$

In this case the a must be chosen to keep $V(\underline{x})$ positive definite and $\dot{V}(\underline{x})$ negative semidefinite. It is convenient to choose the a_i in terms of the system parameters and a_2. By choosing suitable values and thus reducing some terms to zero, the negative semidefiniteness of $\dot{V}$ reduces to a single inequality in a_2 and the system parameters, and a_2 can be chosen to satisfy this inequality. The a_i are chosen as

$$a_1 = \left[\frac{D_a}{2} + \frac{\beta}{4\pi}\right]\frac{\beta}{2\pi T_G} \qquad a_3 = \frac{T_G}{2} \qquad a_4 = T_T$$
$$a_5 = \frac{\beta}{2\pi k_F T_G} \qquad a_6 = \frac{\beta}{2\pi} \qquad a_7 = \frac{\beta T_T}{2\pi T_G} \tag{25}$$
$$a_8 = 2k_F k_L a_2$$

$\dot{V}$ is negative semidefinite if

$$\left(2\, D_a k_F a_2 - \frac{\beta}{2\pi k_F T_G}\right) > \left(k_F a_2 + \frac{\beta T_T}{4\pi T_G}\right)^2 \tag{26}$$

a_2 must satisfy (26) and also make V positive definite. For example, for typical system parameters, a_2 is chosen as

$$a_2 = \frac{1}{16k_F} \tag{27}$$

The critical switching time is found to vary by less than 10% from the maximum obtained by simulation. It is observed that the critical switching time increases as β increases.

GENERATOR WITH FEEDBACK

The system equation is (9) and the Liapunov function is chosen in the same form as (24). As before, $V(\underline{x})$ must be positive definite and $\dot{V}(\underline{x})$ negative semidefinite. The a_i are chosen in terms of a_2 and the system parameters. The following are chosen

$$a_1 = \frac{D_a}{4\pi}\left(K_I - \frac{\beta}{T_G}\right) + \frac{\beta^2}{8\pi^2 T_G} \qquad a_3 = \frac{T_G}{2} \qquad a_4 = T_T$$

$$a_5 = \frac{1}{2\pi k_F}\left(K_I + \frac{\beta}{T_G}\right) \qquad a_6 = \frac{\beta}{2\pi} \qquad a_7 = \frac{T_T}{2\pi}\left(K_I - \frac{\beta}{T_G}\right) \tag{28}$$

$$a_8 = 2k_L k_F a_2$$

$\dot{V}$ is negative semidefinite if

$$\left[2k_F D_a a_2 - \frac{1}{2k_F}\left(K_I + \frac{\beta}{T_G}\right)\right] > \left[k_F a_2 + \frac{T_T}{4\pi}\left(K_I + \frac{\beta}{T_G}\right)\right]^2 \tag{29}$$

a_2 is chosen to satisfy (29) as well as to satisfy the positive definiteness of V.

The boundary of the stability region is no longer as easy to obtain, Since the first unstable equilibrium point, $\underline{x}_u$, encountered as one leaves the origin cannot be obtained by inspection of the equations, one must solve the nonlinear equations numerically, the starting point being the $\underline{x}_u$ of the previous section. The boundary is then obtained as before.

For typical values of the system parameters, a suitable value of a_2 is found to be

$$a_2 = \frac{1}{16k_F} \tag{30}$$

The critical switching time is found to differ from the results obtained by simulation by less then 10%. The critical switching time increases as K_I increases.

CONCLUSION

Liapunov functions have been constructed for a single generator with feedback, and they are found to give good results. However, the conditions under which they are valid Liapunov functions are more restrictive than in the case without feedback and the selection of the constants depends upon the system parameters.

For the examples considered, it is found that the stability improves as β and K_I are individually increased from zero. However, the interaction between these two variables requires more extensive study.

References

1) Lasalle, J. P. & Lefschetz, S -- 'Stability by Liapunov's Direct Method', Academic Press, New York 1961.

2) Gless, G. E. - 'Direct Method of Liapunov Applied to Transient Power System Stability', I.E.E.E. Transactions on P.A.S., vol. 85, Feb. 1966, pp 159-168.

3) Willens, J. L. - 'Direct Methods for Transient Stability Studies in Power System Analysis', I. E. E. E. Transactions on Augomatic Control, Vol. A. C. 16, pp 332-341,Aug. 1971.

4) El-Abiad, A. H. & Nagappan, K. - Transient Stability Regions a Multimachine Power Systems - I. E. E. E. Transactions on P. A. S., Vol 85 Feb. 1966, pp 169-179.

5) Elgerd, O. I. - 'Electric Energy System Theory: An Introduction' - McGraw Hill, New York 1971.

AMBIGUOUS BEHAVIOR OF LOGIC BISTABLE SYSTEMS**

MARCO HURTADO* AND DAVID L. ELLIOTT†

Washington University, St. Louis, Missouri

ABSTRACT

The standard specifications of logic bistable devices do not specify the behavior under conditions in which the input is logically undefined or in which certain kinds of multiple input changes occur. These conditions are unavoidable in logic synchronizers and arbiters. A general deterministic model of bistable devices is proposed, consisting of a nonlinear differential system with some adequate properties. Analysis of this model shows that bistable devices can be driven into a logically undefined region by certain admissible inputs and can remain in this region for an unbounded length of time.

1. INTRODUCTION

This paper is concerned with the behavior of logic bistable elements under input conditions that occur unavoidably in synchronizers that mediate communications between systems that do not share a common time reference, or in asynchronous arbiters that must allocate a resource to one of several users whose competing requests may occur at the same time. The input may be a weak or short pulse, or certain kinds of multiple input changes may occur more or less simultaneously. The resulting misbehavior of these circuits [1] produces serious system control errors. The paper is based on part of the doctoral dissertation written by the first author [2], in which the following concepts are applied to tunnel-diode and transistor flip-flops. Here we consider only a noise-free or deterministic model of a bistable device; in [2] the effect of additive white noise is discussed qualitatively.

In Figures 1 and 2 we have reproduced bistable output waveforms reported in [1]. The pictures at the left have been obtained using a sampling scope (1000dots/division), while the traces at the right are real-time sample responses. The output hangs up at a logically undefined voltage level (MC1016) or oscillates within the logically undefined region under the given input conditions. The time until logical resolution can be much larger than the switching time with ideal inputs.

This behavior is particularly serious when the output of the bistable device must drive, in parallel, subsequent logical circuits; some circuits may be activated but not others, creating a conflict or paradox in the logical system.

Since there is experimental evidence that the larger resolution times are of low probability, the problem may be alleviated by providing a delay time, fixed or depending on a threshold sensing device.

2. MATHEMATICAL MODEL OF A BISTABLE SYSTEM

Our deterministic mathematical model for bistable elements consists of a system of time-invariant nonlinear differential equations driven by a

---

*Research Engineer, Computer Systems Laboratory.

†Associate Professor, Department of Systems Science and Mathematics.

**This investigation was supported in part by NIH Research Grant No. RR0036 from the Biotechnology Resources Branch, by NSF Research Grant No. GK-38694, and by NSF Grant No. GK-22905#3.

piecewise continuous bounded function. It is assumed that the model exhibits the following fundamental properties:

(a) There exists a constant input function for which the system has two asymptotically stable equilibrium points (see, for example, [3], p. 58).

(b) The set of input functions contains a subset whose elements are able to steer the system, in a finite time interval, from a neighborhood of one of the equilibrium points to a neighborhood of the other, and a second subset whose elements are able to steer the system, in the same time interval, in the reverse direction. Those neighborhoods correspond to the "0" and "1" binary logical conditions of the bistable system. They will be referred to as *logical sets*.

In what follows and in the next sections we will be referring often to the concepts and the theorems of dynamical systems theory [3].

In state representation the model is expressed as the differential system

$$\dot{\underline{x}}(t) = \underline{f}[\underline{x}(t),\ \underline{u}(t)] \qquad (1)$$

$$\underline{x}(0) = \underline{\xi}$$

where

(1) $\underline{x}(t) \in R^n$ represents the state of the system at time $t \in [0,T]$.

(2) $\underline{u} = \{\underline{u}(t),\ t \in [0,T]\}$ is an element in the set U of inputs consisting of *m*-vector-valued piecewise-continuous functions defined on the interval [0,T] such that

$$a_i \leq u_i(t) \leq b_i, \quad i=1,2,\dots,m$$

for any $t \in [0,T]$ where the a_i's and b_i's are real numbers. The Cartesian product of the sets $[a_i,b_i]$, $i=1,2,\dots,m$ will be denoted by Ω.

(3) $\underline{f}: R^n \times \Omega \to R^n$ is a continuous mapping satisfying the conditions needed to guarantee that the solution $\psi[t,\underline{\xi},\underline{u}]$ of the system (1) is unique on [0,T] for each $\underline{\xi} \in R^n$ and each $\underline{u} \in U$, and continuous on $[0,T] \times R^n \times U$ in the following sense:

(i) For $\underline{u}$ U, $t,s \in [0,T]$ and $\underline{\xi},\underline{\eta} \in R^n$, given any $\varepsilon > 0$ there exists a $\delta > 0$ such that

$$\sum_{i=1}^{n} |\psi_i[t,\underline{\xi},\underline{u}] - \psi_i[s,\underline{\eta},\underline{u}]| < \varepsilon \text{ whenever } |t-s| + \sum_{i=1}^{n} |\xi_i - \eta_i| < \delta$$

where ξ_i and η_i, $i=1,2,\dots,n$, are the components of $\underline{\xi}$ and $\underline{\eta}$ respectively.

(ii) For $t \in [0,T]$, $\underline{\xi} \in R^n$ and $\underset{\rightarrow}{u}\underline{v} \in U$, given any $\varepsilon > 0$ there exists a $\delta > 0$ such that

$$\sum_{i=1}^{n} |\psi_i[t,\underline{\xi},\underline{u}] - \psi_i[t,\underline{\xi},\underline{v}]| < \varepsilon,$$

whenever

$$\sum_{i=1}^{m} \int_0^t |u_i(t) - v_i(t)|\,dt < \delta .$$

Furthermore, the following assumptions are included in order to satisfy the conditions for bistability.

(4) There exists a constant input function $\{\underline{c}, t \in [0,T]\} \in U$ that, extended in the obvious way to the entire real line R, determines the

autonomous differential equation

$$\dot{\underline{x}}(t) = \underline{f}[\underline{x}(t),\underline{c}] \qquad (2)$$
$$\underline{x}(0) = \underline{\xi}$$

with unique and continuous solutions $\psi_0(t,\underline{\xi})$ on $R\times R^n$ and with two different asymptotically stable equilibrium points in R^n, namely, $\underline{e}_1$ and $\underline{e}_2$. The corresponding regions of attraction ([3],p. 57) are denoted respectively by A_1 and A_2. The system (2) is termed the *autonomous subsystem*.

(5) There exist subsets U_1 and U_2 in U, and neighborhoods $N(\underline{e}_1) \subset A_1$ and $N(\underline{e}_2) \subset A_2$ of $\underline{e}_1$ and $\underline{e}_2$ respectively, such that

$$K[T;\underline{\xi},U_1] \subset A_2$$

and

$$K[T;\underline{\eta},U_2] \subset A_1$$

for any $\underline{\xi} \in N(\underline{e}_1)$ and $\underline{\eta} \in N(\underline{e}_2)$. The set $K[T;\underline{\xi},U_1]$, defined as

$$K[T;\underline{\xi},U_1] = \bigcup_{\underline{u}\in U_1} \{\underline{x}\in R^n: \psi[t,\underline{\xi},\underline{u}] = x\}$$

is called the set of reachable states at T from $\underline{\xi}$ at 0, with respect to U_1. A similar definition corresponds to $K[T;\underline{\eta},U_2]$. The neighborhoods $N(\underline{e}_1)$ and $N(\underline{e}_2)$ are the logical sets.

3. EXISTENCE OF THE REGION OF INDECISEVENESS OF THE AUTONOMOUS SUBSYSTEM

If we define a map $\pi: R^n\times R \to R^n$ such that $\pi(\underline{x},t)=\psi_0(t,\underline{x})$, where $\psi_0(t,\underline{x})$ is the solution of the autonomous subsystem (2), it is clear that (2) is a dynamical system on R^n([3],p. 5). Therefore, the regions of attraction A_1 and A_2 are open invariant neighborhoods of $\underline{e}_1$ and $\underline{e}_2$, respectively ([3], p. 60). Furthermore, the boundary sets ∂A_1 and ∂A_2 of A_1 and A_2, respectively, are also invariant ([3], p. 13). We will show that there exists in R^n a nonempty set initiating solutions that do not approach either $\underline{e}_1$ or $\underline{e}_2$. Let d be any metric compatible with the usual R^n topology.

Proposition 1. The boundary sets ∂A_1 and ∂A_2 of A_1 and A_2, respectively, are nonempty closed sets.

Proof: The regions of attraction, A_1 and A_2, are disjoint since, otherwise, there exists a $\underline{x}\in R^n$ such that $\underline{x}\in A_1$ and $\underline{x}\in A_2$ and consequently $d[\psi_0(t,\underline{x}),\underline{e}_1] \to 0$ and $d[\psi_0(t_1\underline{x}),\underline{e}_2] \to 0$ as $t \to \infty$. This implies that $\underline{e}_1=\underline{e}_2$, which contradicts the assumption that $\underline{e}_1$ and $\underline{e}_2$ are different. Therefore, $A_1\cap A_2=\phi$. Furthermore, $A_1\cup A_2\neq R^n$ since A_1 and A_2 are open sets. Therefore ∂A_1 and ∂A_2 are nonempty. They are also closed by definition of a boundary set.

Proposition 2. ∂A_1 and ∂A_2 are unstable sets.

Proof: The proof is presented for A_1. Since A_1 is an open neighborhood of $\underline{e}_1$ in R^n, then $A_1 \cap \partial A_1=\phi$, and $d[\partial A_1,\underline{e}_1] > 0$. Let $S(\partial A_1,\varepsilon)$ denote the ε-neighborhood of ∂A_1 defined as the set

$$\{\underline{x}\in R^n: d[\underline{x},\underline{y}] < \varepsilon,\ \underline{y} \in \partial A_1\}$$

where $\varepsilon \in R^+$. We now choose a $\varepsilon \in R$ such that $d[S(\partial A_1,\varepsilon),\underline{e}_1] > 0$ and assume that ∂A_1 is stable ([3],p. 84). Then, for any $\underline{z} \in \partial A_1$ there exists a $\gamma(\underline{z},\varepsilon)\in R$, $\gamma(\underline{z},\varepsilon) > 0$, such that the positive semitrajectories of the γ-neighborhood of $\underline{z}$ are in $S(\partial A_1,\varepsilon)$; that is

$$\{\underline{x} \in R^n: \psi_0(\underline{y},t) = \underline{x},\ \underline{y}\in S(\underline{z},\gamma),\ t\in R^+\} \subset S(\partial A_1,\varepsilon).$$

But, $\underline{z}$ being a point in ∂A_1, its γ-neighborhood $S(z,\varepsilon)$ must have at least a point in A_1, say the point $\underline{w}$. Consequently, $d[\psi_0(\underline{w},t),\underline{e}_1] \to 0$ as $t \to \infty$.

Therefore

$$\{\underline{x} \in R^n:\ \psi_0(\underline{w},t) = \underline{x},\ t \in R^+\} \not\subset S(\partial A_1,\varepsilon),$$

which is a contradiction. Therefore A_1 is unstable.

Proposition 3. Given $\underline{x} \in A_1$, $\underline{x} \neq \underline{e}_1$, then the negative limit set, $L^-(\underline{x})$ ([3], p. 19), is included in the boundary set ∂A_1.

Proof: By a theorem ([3], p. 90) of dynamical system theory, the first negative prolongation limit set of $\underline{x}$, $J^-(\underline{x})$ ([3], p. 24), and A_1 are disjoint since $\underline{x} \in A_1 - \underline{e}_1$. Therefore $L^-(\underline{x}) \cap A_1 = \phi$ since $L^-(\underline{x}) \subset J^-(\underline{x})$. Let $\underline{y} \in L^-(\underline{x})$; then there exists a sequence $\{t_n\} \in R$ with $t_n \to -\infty$, such that $\psi_0(\underline{x},t_n) \to \underline{y}$. The sequence $\{\psi_0(\underline{x},t_n):\ n=1,2,\ldots\}$ is in A_1 since A_1 is an invariant set. Consequently $\underline{y}$ is in the closure $\bar{A}_1$ of A_1 and since $\underline{y}$ is not in A_1, then $\underline{y} \in \partial A_1$.

A similar proposition applies to the negative limit set of a point, other than $\underline{e}_2$, in the region of attraction A_2.

Proposition 4. Let $\partial A_1 \neq \partial A_2$. Then

(i) If $\partial A_1 \cap \partial A_2 \neq \phi$, $\partial A_1 \cap \partial A_2$ is a closed invariant set.
(ii) If $\partial A_1 \cap \partial A_2$ is a singleton, then it is an unstable equilibrium point.
(iii) If $\bar{A}_1$ and $\bar{A}_2$ are the closures of, respectively, A_1 and A_2, then $R^n - (\bar{A}_1 \cup \bar{A}_2)$ is also an invariant set.

Proof: (i) $\partial A_1 \cap \partial A_2$ is a closed invariant set because ∂A_1 and ∂A_2 are closed and invariant ([3], p. 12).

(ii) Let $\{\underline{x}\} = \partial A_1 \cap \partial A_2$; then $\psi_0(\underline{x},t) = \underline{x}$ for every $t \in R$ because of (i). Thus $\underline{x}$ is a critical point by definition. It is unstable since ∂A_1 and ∂A_2 are unstable because of Proposition 2.

(iii) $\bar{A}_1$, $\bar{A}_2$ and $\bar{A}_1 \cup \bar{A}_2$ are invariant ([3], p. 12). Therefore, $R^n - (\bar{A}_1 \cup \bar{A}_2)$ is also invariant ([3], p. 13).

Comments: (1) The propositions of this section can be extended in a straightforward manner to a case of autonomous dynamical system with more than two asymptotically stable equilibrium points.

(2) Since ∂A_1, ∂A_2 and $R^n - (\bar{A}_1 \cup \bar{A}_2)$ are invariant, the set $R^n - (A_1 \cup A_2)$ initiates trajectories of the autonomous subsystem not approaching either one of the asymptotically stable equilibrium points. This set is called the *region of indecisiveness*.

(3) If the region of indecisiveness is composed of only the boundary sets of the two regions of attraction, it is unstable.

(4) Trajectories of the autonomous subsystem starting in a region of attraction of one of the desired equilibrium points may require an arbitrarily long time to reach the logical set within such a region of attraction, since the boundary of a region of attraction is the negative limit set of the points in the region.

4. EXISTENCE OF ADMISSIBLE INPUTS
DETERMINING AMBIGUOUS BEHAVIOR

Assuming that the bistable system state is initially at a point in the logical set $N(\underline{e}_1)$ $[N(\underline{e}_2)]$, and the input function applied in the interval $[0,T]$ belongs to the subset $U_1[U_2]$ and is followed by the constant input $\underline{c}$, the system state will change from the initial logical condition to the other, within the finite interval of duration T, and will permanently remain in this second condition. But if the input function is one that would bring the state of the system at T to a point in the region of indecisiveness, after T the state would remain in this region so that the system will not ever achieve a logical definition. In this section we show that there are indeed admissible input functions that will produce a permanent logically undefined condition of the bistable system.

We now consider the metric space composed by the input-function set, U, and the metric

$$d^1[\underline{u},\underline{v}] = \sum_{i=1}^{m} \int_0^T |u_i(t)-v_i(t)|dt \quad .$$

For the sake of simplicity of notation we will denote it also by U.

Proposition 5. The metric space U is connected.

Proof: Let $\underline{u}$ and $\underline{v}$ be any two elements in U. Define the mapping $\underline{h}$: $[0,1] \to U$ defined by

$$\underline{h}(s,t) = s\underline{u}(t) + (1-s)\underline{v}(t), \quad s \in [0,1]$$

Obviously $\underline{h}(0,t)=\underline{v}(t)$ and $\underline{h}(1,t)=\underline{u}(t)$, and for $s \in [0,1]$, $\underline{h}(s,\cdot)$ is in U. Furthermore, for given s and r in (0,1), we have

$$d^1[\underline{h}(x,\cdot),\underline{h}(r,\cdot)] = \sum_{i=1}^{m} \int_0^T |h_i(s,t)-h_i(r,t)|dt = |s-r| \; d^1[\underline{u},\underline{v}]$$

so that for a given $\varepsilon > 0$, there is a $\delta(\varepsilon) = \varepsilon/(d^1[\underline{u},\underline{v}])$ such that $|s-r| < \delta(\varepsilon)$ implies $d^1[\underline{h}(s,\cdot), h(r,\cdot)] < \varepsilon$. Also $\underline{h}(s,\cdot) \to \underline{v}$ as $s \to 0$ and $\underline{h}(s,\cdot) \to \underline{u}$ as $s \to 1$. Consequently the mapping $\underline{h}$ is continuous on [0,1], and the metric space U is arcwise connected and therefore connected.

Given an $\underline{\xi} \in R^n$ and a $T \in R$ we now define a mapping $\psi_{T,\underline{\xi}}$: $U \to R^n$ such that for a $\underline{u}$ in U $\psi_{T,\underline{\xi}}[\underline{u}] = \psi[T,\underline{\xi},\underline{u}]$, the solution of the bistable system (1). The image $\psi_{T,\underline{\xi}}[U]$ of the metric space U with respect to the mapping $\psi_{T,\underline{\xi}}$ is connected and further, arcwise connected, since the mapping is continuous. The connectivity of $\psi_{T,\underline{\xi}}(U)$ is used in the next proposition. Notice that $\psi_{T,\underline{\xi}}(U)$ is the set of reachable states at T from $\underline{\xi}$ at 0, with respect to U, that is, $\psi_{T,\underline{\xi}}(U) = K[T;\underline{\xi},U]$.

Proposition 6. If the state of the bistable system is in either $N(\underline{e}_1)$ $N(\underline{e}_2)$ at time 0, then there exist input functions in U that bring the state to the boundary sets of the regions of attraction A_1 and A_2 at time T.

Proof: We are going to prove the proposition for the set $N(\underline{e}_1)$. Let the state at time 0 be $\underline{\xi} \in N(\underline{e}_1)$. If the constant input function $\underline{c}$ in U is applied, the state of the system will remain in A_1 during all the time of application of such function, since $N(\underline{e}_1)$ in in A_1 and this set is invariant whenever $\underline{c}$ is applied. Thus $\psi_{T,\underline{\xi}}[\underline{c}] \subset A_1$. On the other hand, for $\underline{u}$ in U_1, $\psi_{T,\underline{\xi}}[\underline{u}]$ is in A_2, because of assumption (5) in Section 2. Therefore, we have in U elements $\underline{c}$ and $\underline{u}$ whose images due to $\psi_{T,\underline{\xi}}$ are in A_1 and A_2, respectively. Therefore, $\psi_{T,\underline{\xi}}[U] \cap A_1 \neq \phi$ and $\psi_{T,\underline{\xi}}[U] \cap R^n - A_1 \neq \phi$. Then, $\psi_{T,\underline{\xi}}[U] \cap \partial A_1 \neq \phi$ ([4], Thm. 3.19.9). That is, there are some $\underline{u} \in U$ such that $\psi_{T,\underline{\xi}}[\underline{u}] \in \partial A_1$. Similarly, $\psi_{T,\underline{\xi}}[U] \cap A_2 \neq \phi$ and $\psi_{T,\underline{\xi}}[U] \cap R^n - A_2 \neq \phi$; therefore, $\psi_{T,\underline{\xi}}[U] \cap \partial A_2 \neq \phi$. That is, there are some $\underline{v} \in U$ such that $\psi_{T,\underline{\xi}}[\underline{v}] \in \partial A_2$.

Proposition 7. Let the state of the bistable system at time 0 be in either $N(\underline{e}_1)$ or $N(\underline{e}_2)$. Assume that the closures $\bar{A}_1$ and $\bar{A}_2$ of the regions of attraction are disjoint; then there are input functions in U that bring the state to the subset $R^n - (\bar{A}_1 \cup \bar{A}_2\ )$ of R^n at time T.

Proof: We are going to prove the proposition for $N(\underline{e}_1)$. Let $\underline{\xi} \in N(\underline{e}_1)$ be the system state at time 0, and E represent the set $\psi_{T,\underline{\xi}}[U]$. From the proof of the last proposition, we have $E \cap \bar{A}_1 \neq \phi$ and $E \cap \bar{A}_2 = \phi$. Now suppose that $E \cap (R^n - (\bar{A}_1 \cup \bar{A}_2)) = \phi$ so that $E = (E \cap \bar{A}_1) \cup (E \cap \bar{A}_2)$. But $E \cap \bar{A}_1$ and $E \cap \bar{A}_2$ are closed in E and disjoint. Consequently, E is not a connected set, which is a contradiction of what was found before. Therefore E or $\psi_{T,\underline{\xi}}[U]$ intersects $R^n - (\bar{A}_1 \cup \bar{A}_2)$ in a nonempty set. In turn this implies that there are some $\underline{u} \in U$ such that $\psi_{T,\underline{\xi}}[\underline{u}]$ is in $R^n - (\bar{A}_1 \cup \bar{A}_2)$.

5. COMMENTS AND CONCLUSIONS

In electronic circuits, in general, there is always present inherent noise. The reference [2] includes a discussion on the noise effect on the operation of bistable devices. Due to the noise, and because of the instability of either the region of indecisiveness or subsets of it, the state of bistable devices, when driven by certain admissible input functions, stay in a logically undefined condition for an unbounded length of time rather than forever.

Based on the results of the paper, we conclude that it is not possible to expect with certainty the achievement of a logical condition by bistable devices within a fixed time interval, when the allowed input signals are bounded but otherwise unrestricted.

6. ACKNOWLEDGEMENT

We wish to acknowledge the invaluable help provided by Professor Charles E. Molnar, advisor for the dissertation on bistable dynamical systems, whose results are partially reported in this paper.

REFERENCES

[1] T.J. Chaney and C.E. Molnar, "Anamalous Behavior of Synchronizer and Arbiter Circuits", IEEE Transactions on Computers, Vol. C-22, No. 4, pp. 421-422, April 1973.

[2] M. Hurtado, "Structure and Performance of Asymptotically Bistable Dynamical Systems", Doctoral Dissertation, Sever Institute of Technology, Washington University, St. Louis, Missouri, May 1975.

[3] N.P. Bhatia and G.P. Szegö, "Stability Theory of Dynamical Systems", Springer Verlag, Berlin, 1970.

[4] J. Dieudonné, "Foundations of Modern Analysis", Academic Press, New York, p. 66, 1960.

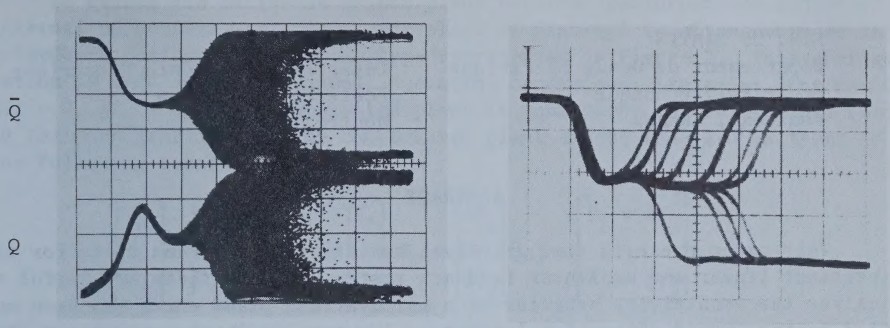


FIGURE 1. Sampling and real-time oscilloscope displays of the response of an ECL clocked R-S flip-flop (Motorola MC1016) to the clock input signal being switched off as the data input signal is changing. 5 nsec./div., 0.25V/div. for sampling displays; 10 nsec./div., 0.2V/div. for real-time displays.

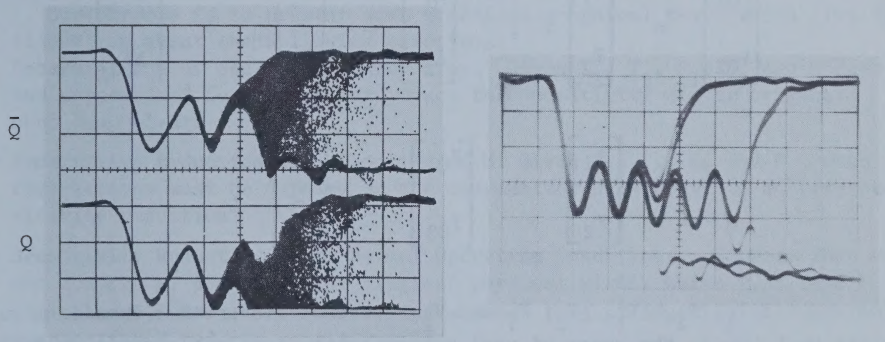


FIGURE 2. Sampling and real-time oscilloscope displays of the response of a TTL R-S flip-flop (constructed by cross-tying two SN7400 NAND gates) to both inputs being changed HIGH simultaneously (sampling displays) and to a weak pulse applied to one input (real-time displays). 5 nsec./div., 1V/div. for sampling displays; 10 nsec./div., 1V/div. for real-time displays.

PRACTICAL SENSITIVITY REDUCTION FOR LINEAR AND NONLINEAR SYSTEMS*

R.M. DE SANTIS
Département de Génie Electrique
Ecole Polytechnique
Montréal, Canada

JEAN CONAN
Département de Génie Electrique
Ecole Polytechnique
Montréal, Canada

ABSTRACT

This paper presents some graphical sensitivity reduction tests for time invariant linear and nonlinear feedback systems. These tests are useful to analyze the sensitivity behavior of a multivariate plant which has been modified by the introduction of a 1-input - 1-output feedback compensator. They are based on Nyquist plot and describing function techniques and involve little extra effort in addition to that which is usually required to determine stability. The modalities of application are illustrated via the analysis of the effect of a frequency-voltage intercoupling in the control system of an isolated hydroelectric power plant.

INTRODUCTION

We will consider the 1 input -2output linear plant represented in Figure 1. This plant will be studied in terms of the following mathematical model

$$\begin{bmatrix} y_{01} \\ y_{02} \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} u \; ; \quad \begin{bmatrix} \overset{0}{\Delta}_1 \\ \overset{0}{\Delta}_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \eta + \begin{bmatrix} \Delta T_1 \\ \Delta T_2 \end{bmatrix} u$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} y_{01} \\ y_{02} \end{bmatrix} + \begin{bmatrix} \overset{0}{\Delta}_1 \\ \overset{0}{\Delta}_2 \end{bmatrix} .$$

The symbols T_1, T_2, ΔT_1, ΔT_2, F_1 and F_2 represent linear time invariant mappings on $L_2[0,\infty)$, the space of real valued square integrable function defined on the real line; y_{01}, y_{02}, u, η, $\overset{0}{\Delta}_1$, $\overset{0}{\Delta}_2$, y_1 and y_2 are elements of $L_2[0,\infty)$. T_1 and T_2 describe the input-output behavior of the plant under nominal operating conditions; ΔT_1 and ΔT_2 take into account the variation of the plant parameters from their nominal values; F_1 and F_2 are used to evaluate the influence of an external perturbation. Finally, the elements y_{01}, y_{02}, u, η, $\overset{0}{\Delta}_1$, $\overset{0}{\Delta}_2$, y_1 and y_2 represent respectively the nominal output, the input, the external perturbation, the output error and the output of the plant.

---

* This research was supported in part by the grant CNRC-A8244

To reduce the influence of the plant external perturbations and/or of internal parameters variations, the plant of Figure 1 is often embedded in a feedback configuration of the type represented in Figure 2. This configuration is such that under normal operating conditions, (that is: for $\eta = 0$, $\Delta T_1 = 0$, $\Delta T_2 = 0$), the compensated plant is equivalent to the original one; in the more general case the compensated plant can be modelled in terms of the following equations

$$\begin{bmatrix} y_{01} \\ y_{02} \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} u \; ; \; \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \eta + \begin{bmatrix} \Delta T_1 \\ \Delta T_2 \end{bmatrix} [u] - \begin{bmatrix} T_1 + \Delta T_1 \\ T_2 + \Delta T_2 \end{bmatrix} M(y^*_1 - y_{01})$$

$$\begin{bmatrix} y^*_1 \\ y^*_2 \end{bmatrix} = \begin{bmatrix} y_{01} \\ y_{02} \end{bmatrix} + \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix}.$$

We will say that the compensated plant reduces the sensitivity of the first, (second), output if the first, (second), output error in the compensated plant is smaller than that in the original plant, that is if

$$\frac{\| \Delta_1 \|}{\| \Delta_1^0 \|} < 1, \quad \left(\frac{\| \Delta_2 \|}{\| \Delta_2^0 \|} < 1 \right).$$

Our purpose is to present some practical graphical tests which simplify a discussion about the following questions:

- Determination of Sensitivity Reduction: Given T_1, T_2, F_1, F_2 and M does the compensated feedback plant reduce the sensitivity of the original open loop plant?
- Sensitivity Reduction as a function of M: Given T_1, T_2, F_1 and F_2, what restrictions must be imposed on the compensator M in order to achieve sensitivity reduction?
- Sensitivity Reduction under Normal Operating Conditions: Suppose that we can observe Δ_1^0 and Δ_2^0 in the original physical plant, while η, F_1 and F_2 are unknown. How can one determine whether the introduction of a given compensator M reduces sensitivity?

The development will first consider the case where M is linear and time invariant. Later we will also discuss the case where M is given by the cascade composition of a linear operator M' followed by a nonlinear memoryless operator N. The general multivariable nonlinear and time variant case is dicussed in [3].

Example (Cuenod and Alias [6]). The influence of small load variations over voltage and frequency deviations in the isolated hydroelectric power plant represented in Figure 3, is usually studied in terms of the linear time invariant model represented in Figure 4; the effect of introducing a frequency-voltage regulator can be analyzed in terms of the compensated model in Figure 5. Denoting the load variation as η, the frequency and voltage deviations respectively as Δ_1^0, Δ_1, Δ_2^0 and Δ_2, and the frequency voltage regulator as M, one recognizes that these two models are of the type represented in

Figures 1 and 2. It follows that the engineering question about whether the introduction of a frequency-voltage regulator improves or deteriorates the power plant performance can be studied in terms of the mathematical questions considered in this development.

MAIN RESULTS

The analysis of the sensitivity questions under consideration is in general quite complex and usually calls for the utilization of advanced mathematical techniques [1], [2], [3]. Our graphical tests however turn out to be very simple. In the linear case, this simplicity follows from the assumption that the perturbation is represented by a sinusoidal function. In the nonlinear case we will further assume that our system satisfies those reasonable engineering conditions under which the describing function approach is usually applied (see [4], chapter 6). In this vein the errors $\Delta_1(t)$ and $\Delta_2(t)$ will be approximated in terms of their first harmonics $\tilde{\Delta}_1(t)$ and $\tilde{\Delta}_2(t)$. The following simplified sensitivity reduction criterion will be used: the compensated system reduces the sensitivity of the i-th output of the original system, if the amplitude of $\tilde{\Delta}_i(t)$ is smaller than the amplitude of $\Delta_i^0(t)$.

A direct approach to the analysis of sensitivity reduction would suggest the utilization of a setting of the type represented in Figure 6. In this setting η and u are applied to both the compensated and the open loop plants; the outputs of these plants are substracted from the output of the nominal plant. This makes it possible to compare Δ_1^0, Δ_2^0, with Δ_1 and Δ_2 in correspondence with various values of η, ΔT_1, ΔT_2, u and M. Though not very efficient, this approach would in principle allow one to answer the question about whether sensitivity reduction has been attained or not. To do this, however, both u and η are required to be accessible; a condition which is not usually satisfied. Thus the problem of determining sensitivity reduction under normal operating conditions remains unsolved. An additional and more relevant shortcoming of the direct approach is that it does not offer the insight which is needed to develop a well defined compensator's design strategy leading to sensitivity reduction. The following propositions alleviate in part these two shortcomings, and are at the basis of the alternative approach presented in the subsequent sections.

Proposition 1. The closed and open loop errors Δ_1, Δ_2 and Δ_1^0, Δ_2^0 can be obtained the one from the other by using the sensitivity models represented in Figures 7 and 8.

Proposition 2. Using the notation $\tilde{T}_i = T_i + \Delta T_i$, $i = 1,2$, the sensitivity of the first output of the compensated plant is better than that of the original plant if $\|I + \tilde{T}_1 M\| > 1$. The sensitivity of the second output is better if

$$\|I - T_2 M (I + T_1 M)^{-1} F_1 F_2^{-1}\| < 1, \quad \text{when } \Delta T_1 = \Delta T_2 = 0$$

or if

$$\|I - \tilde{T}_2 M (I + \tilde{T}_1 M)^{-1} \Delta T_1 \, \Delta T_2^{-1}\| < 1, \quad \text{when } \eta = 0.$$

If we suppose that the mappings T_1, T_2, F_1, F_2 and M are time invariant and that $\Delta T_1 = \Delta T_2 = 0$, then from Proposition 2 we obtain the following results,

Corollary 1. (Nyquist conditions for a given M). A sufficient condition for the sensitivity reduction of the first, (second), output is that the Nyquist plot of T_1M, $(T_2M(I + T_1M)^{-1} F_1F_2^{-1})$, be exterior, (interior), to the unit circle with center at the point $(-1,0)$, $((1,0))$.

Corollary 2. (Nyquist conditions as an explicit function of M). A sufficient condition for the sensitivity reduction of the first, (second), output is that the origin, (the point $P_2 = [T_1 - T_2F_1F_2^{-1}]\,(j\omega)$) be interior to the circle which has center at $-1/M(j\omega)$ and which passes by $T_1(j\omega)$. The ratios

$$\frac{\| \Delta_1^{\circ} \|}{\| \Delta_1 \|} \quad \text{et} \quad \frac{\| \Delta_2^{\circ} \|}{\| \Delta_2 \|}$$

can be evaluated graphically using the following expressions:

$$\left| \frac{\Delta_1^{\circ}}{\Delta_1}(j\omega) \right| = \left| \frac{1}{M(j\omega)} + T_1(j\omega) \right| \Big/ \left| \frac{1}{M(j\omega)} \right| ; \left| \frac{\Delta_2^{\circ}}{\Delta_2}(j\omega) \right| = \left| \frac{1}{M(j\omega)} + T_1(j\omega) \right| \Big/ \left| \frac{1}{M(j\omega)} + P_2(j\omega) \right.$$

Corollary 3. (Nyquist conditions for the closed loop system configuration). A sufficient condition for the sensitivity reduction of the first, (second), output is that $T_1M(I + T_1M)^{-1}\,(j\omega)$, $(T_2M(I + T_1M)^{-1}\,(j\omega))$, is interior to the circle which passes through the origin and has center at $(1,0)$, $(F_2F_1^{-1}(j\omega))$.

Proposition 3. By implementing the following change of notations

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \rightarrow \begin{bmatrix} \Delta T_1 \\ \Delta T_2 \end{bmatrix}$$

the above corollaries are applicable to the case where $T_1 = 0$ and ΔT_1 and ΔT_2 are different from zero.

APPLICATION

Suppose that the assigned plant P (Figure 1), is embedded in a feedback configuration adopting a specific compensator M (Figure 2). How can we verify whether this embedding reduces sensitivity? Using Corollary 2, this question can be answered by implementing the following procedure:

Step 1: Determine the frequency response of T_1, T_2, F_1 and F_2;
Step 2: Draw the Nyquist plots of T_1, $P_2 = T_1 - T_2F_1F_2^{-1}$, and $-1/M$;
Step 3: Discuss sensitivity reduction in terms of the graphical determination of

$$\left| \frac{\Delta_1^{0}}{\Delta_1}(j\omega) \right| \quad \text{and} \quad \left| \frac{\Delta_2^{0}}{\Delta_2}(j\omega) \right|$$

from the Nyquist plots of T_1, P_2 and $-1/M$.

We will now apply the above procedure to discuss the effect of introducing a linear frequency-voltage intercoupling over voltage and frequency sensitivity to small load variations in the hydroelectric plant represented in Figure 3.

Step 1: Using Figure 4, and adopting the nominal values proposed in [6], we have

$$T_1(j\omega) = \frac{-4j\omega}{[16(j\omega)^3 + 12(j\omega)^2 + 2j\omega + 1]}$$

$$T_2(j\omega) = \frac{[16(j\omega)^3 + 8(j\omega)^2 + 2j\omega + 1]}{[16(j\omega)^3 + 12(j\omega)^2 + 2j\omega + 1][j\omega + 1]}$$

$$F_1(j\omega) = \frac{2j\omega[(j\omega)^3 + 2.2(j\omega)^2 + j\omega + 1]}{[16(j\omega)^3 + 12(j\omega)^2 + 2j\omega + 1][j\omega + 1]^2}$$

$$F_2(j\omega) = \frac{j\omega[8.4(j\omega)^4 + 25.2(j\omega)^3 + 14.8(j\omega)^2 + 4.4j\omega + 1]}{[16(j\omega)^3 + 12j\omega + 2s + 1][j\omega + 1]^3}$$

Step 2: Suppose that M is given by a PD compensator; in particular let $M(j\omega) = -1 - 3.8j\omega$. The Nyquist plot of $-\frac{1}{M(j\omega)}$, $T_1(j\omega)$ and $P_2(j\omega)$ are sketched in Figure 9.

Step 3: From the Nyquist plots of T_1, P_2 and $-1/M$ we can see that the ratios

$$\left|\frac{\Delta_1^0(j\omega)}{\Delta_1(j\omega)}\right| \quad \text{and} \quad \left|\frac{\Delta_2^0(j\omega)}{\Delta_2(j\omega)}\right|$$

are smaller than 1 for some values of ω and larger than 1 for other values. Thus for example for $\omega = .05 \times 2\pi$ we have

$$\frac{\Delta_1}{\Delta_1^o} = \frac{m_1}{m_2} \cong .1 \text{ and } \frac{\Delta_2}{\Delta_2^o} = \frac{m_3}{m_2} \cong .59.$$

This means that if the load variation is sinusoidal with a frequency $f = .05c/s$, then the proposed frequency-voltage intercoupling reduces the frequency error by a 20dB factor; while the voltage error is reduced by a 4.6 dB factor.

Note, however, that for $\omega = .03 \times 2\pi$ we have

$$\frac{\Delta_1}{\Delta_1^o} = \frac{m_1'}{m_2'} \cong .58 \text{ and } \frac{\Delta_2}{\Delta_2^o} = \frac{m_3'}{m_2'} \cong 2.24.$$

This means that while for a sinusoidal load variation of frequency $f \cong .03c/s$ the proposed intercoupling reduces the frequency error by 4.7dB, the voltage error is increased by 7dB.

THE CASE OF THE NONLINEAR COMPENSATOR

In this section we will suppose that the feedback compensator appearing in Figure 2 is given by the cascade composition of a nonlinear memoryless and time invariant system, N, followed by a linear and time invariant system M'. This case is more involved than that considered in section 3 because now for a sinusoidal perturbation of a given frequency, the compensated plant may have a smaller sensitivity than the original plant for some values of the amplitude and a larger sensitivity for other amplitude values. The test which distinguishes these two possible occurences is established by the following proposition. In stating this proposition, the symbol $\tilde{N}(.)$ will denote the describing function associated with the nonlinearity N; the symbol δ_1 will represent the amplitude of the first harmonic of the error function $\Delta_1(t)$.

Proposition 4. For a sinusoidal perturbation of amplitude A and angular velocity ω_o, the compensated plant reduces the sensitivity of the i-th output if A and ω_o are such that $\bar{P}_i(j\omega)$ is interior to the circle which has center at $-\frac{1}{N(\delta_1)}$ and passes through the point $M'T_1(j\omega)$, where $\bar{P}_i(j\omega) = 0$ and $\bar{P}_2(j\omega) = M'(T_1F_2 - T_2F_1)F_2^{-1}(j\omega)$. A proof of this proposition together with an illustrative application example can be found in [8].

CLOSURE

The main results of this paper permit to analyze the sensitivity improvement which can be obtained by introducing a 1 input - 1 output linear or nonlinear feedback compensator into a multivariate plant. In the linear case, Proposition 2 gives a simple procedure to establish sensitivity performance in terms of Nyquist plots conditions. Proposition 4.1 suggests that by considering the critical locus associated with the nonlinearity a similar procedure can be developed for the nonlinear case. These two procedures can both be implemented graphically and require little extra effort in addition to that which is usually undertaken to verify stability conditions with the Nyquist and describing function criteria [7], [4].

REFERENCES

[1] Benchmark Papers in System Sensitivity Analysis, J.B. Cruz, Jr., (Editor), Dowdon, Hutchison & Ross, Inc., 1973.

[2] General Sensitivity Theory, R. Tomovic and M. Vukobratovitch, American Elsevier, New-York, 1972, See also review, IEEE Trans. Auto. Control, Vol. AC-18, No. 2, April 1973, W.A. Porter.

[3] A Generalized Nyquist plot and its Use in Sensitivity Analysis, R.M. De Santis, W.A. Porter, International Journal on System Sciences.

[4] Modern Control Principles and Applications, J.C. Hsu, and A.V. Mayer, McGraw-Hill, 1968.

[5] Circle type conditions for Sensitivity Reduction, R.M. De Santis, W.A. Porter, ACTA, Vol. 2, No.2, May 1974.

[6] Règlage de Tension et de Fréquence dans un Groupe Hydroélectrique, B. Dubé, M. Cuenod, et alias, ACTA, Vol. 1, No. 1, January 1972.

[7] Diagramme de Nyquist Inverse et Stabilité d'un Système Asservi, Rapport technique de l'Ecole Polytechnique de Montréal, EP-75-R-27, Q.T. Vu, R.M. De Santis, May 1975.

[8] Practical Sensitivity Reduction Tests for Linear and Nonlinear Systems, Rapport technique de l'Ecole Polytechnique de Montréal, EP-74-R-28, R.M. De Santis, May 1974.

[9] Règlage Optimal de Tension et Fréquence dans une Centrale Hydroelectrique, J. Conan, R.M. De Santis, ACFAS, Mai 1975.

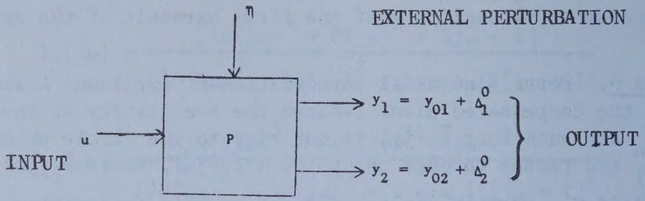

η
EXTERNAL PERTURBATION
u
INPUT
P
$y_1 = y_{01} + \Delta_1^0$
$y_2 = y_{02} + \Delta_2^0$
OUTPUT

Figure 1: The open loop plant.

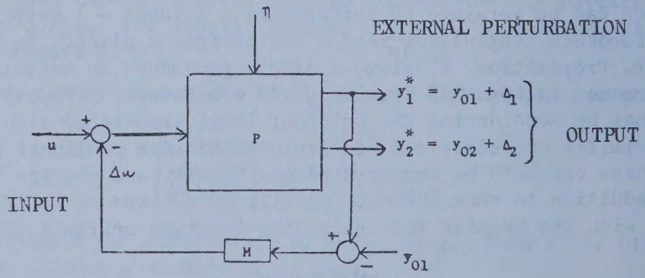

η
EXTERNAL PERTURBATION
+
u
-
Δω
INPUT
P
$y_1^* = y_{01} + \Delta_1$
$y_2^* = y_{02} + \Delta_2$
OUTPUT
M
+
-
y_{01}

Figure 2: The compensated plant.

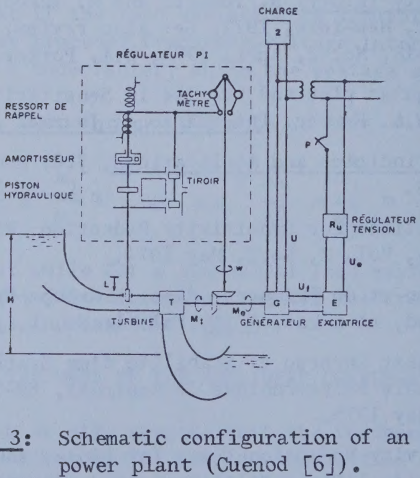

CHARGE
2
RÉGULATEUR PI
RESSORT DE RAPPEL
TACHYMÈTRE
AMORTISSEUR
PISTON HYDRAULIQUE
TIROIR
P
Ru
RÉGULATEUR TENSION
U
Ue
W
Uf
L
H
T
TURBINE
Mt
Me
G
GÉNÉRATEUR
E
EXCITATRICE

Figure 3: Schematic configuration of an isolated hydroelectric power plant (Cuenod [6]).

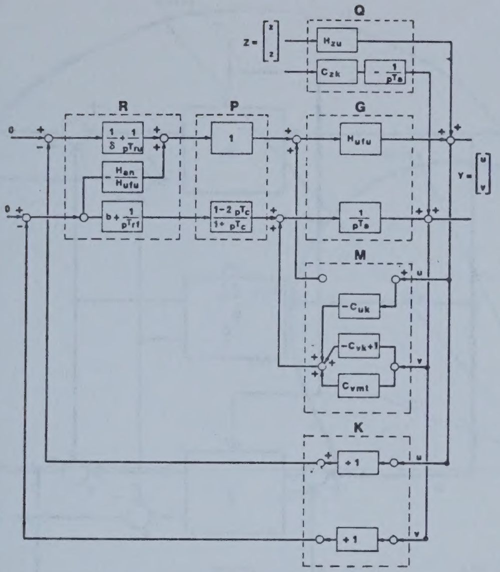

Q
Z =
z
z
Hzu
Czk
- 1/pTa
R
P
G
0
0
1/δ + 1/pTru
- Han/Hufu
b + 1/pTrf
1
1−2 pTc / 1+ pTc
Hufu
1/pTa
Y =
u
v
M
−Cuk
−Cvk+1
Cvmt
K
+1
+1
u
v

Figure 4: Linear model associated with the isolated hydroelectric power plant (Cuenod [6]).

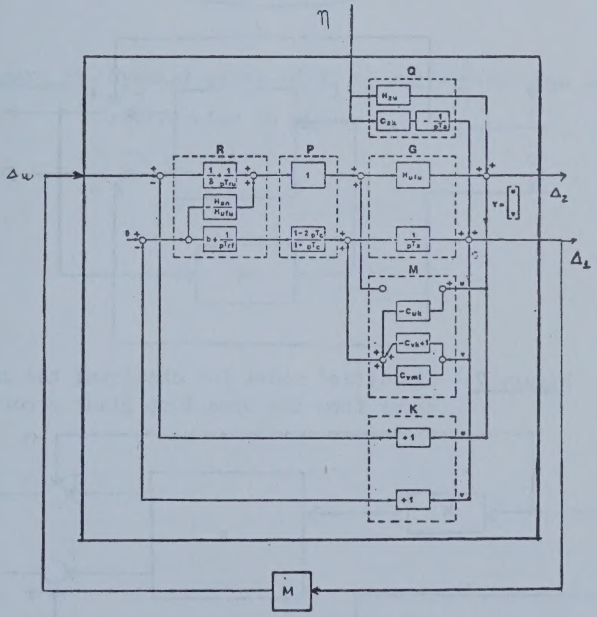
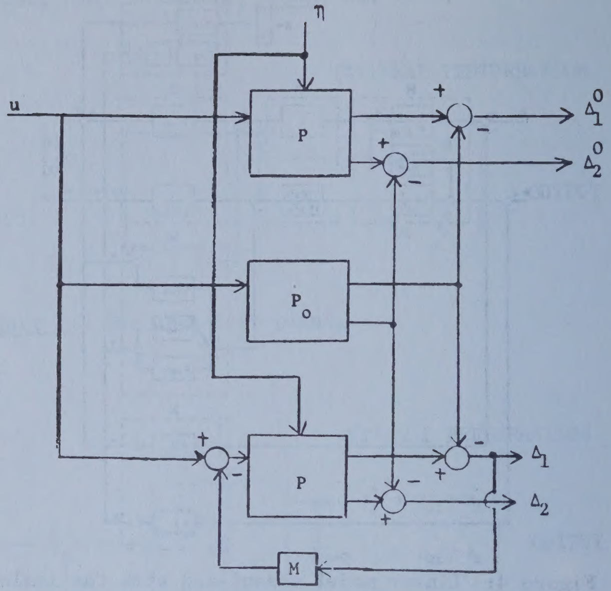
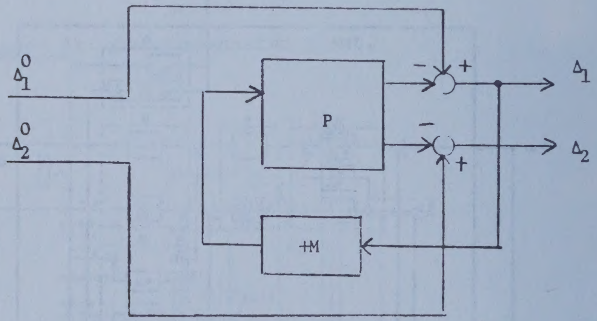

η
Δω
Δz
Δ⊥
Q
Hzu
Czk
- 1/pTa
R
P
G
1/δ + 1/pTru
- Han/Hufu
b + 1/pTrf
1
1−2 pTc / 1+ pTc
Hufu
1/pTa
Y =
u
v
M
−Cuk
−Cvk+1
Cvmt
K
+1
+1
M

Figure 5: Representation of the Intercoupling between frequency and voltage loops in the hydroelectric plant of Figure 3 (Cuenod [6]).



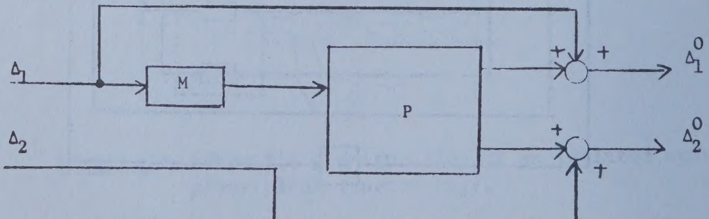
η
u
P
+
−
Δ_1^0
Δ_2^0
P_o
P
M
Δ_1
Δ_2

Figure 6: Setting used in the direct approach to sensitivity reduction (P_o represents the nominal plant).



Δ_1^0
Δ_2^0
P
+M
−
+
Δ_1
Δ_2

Figure 7: Simplified model for obtaining the compensated plant error from the open loop plant error.



Δ_1
Δ_2
M
P
+
Δ_1^0
Δ_2^0

Figure 8: Simplified model for obtaining the open loop plant error from the compensated plant error.

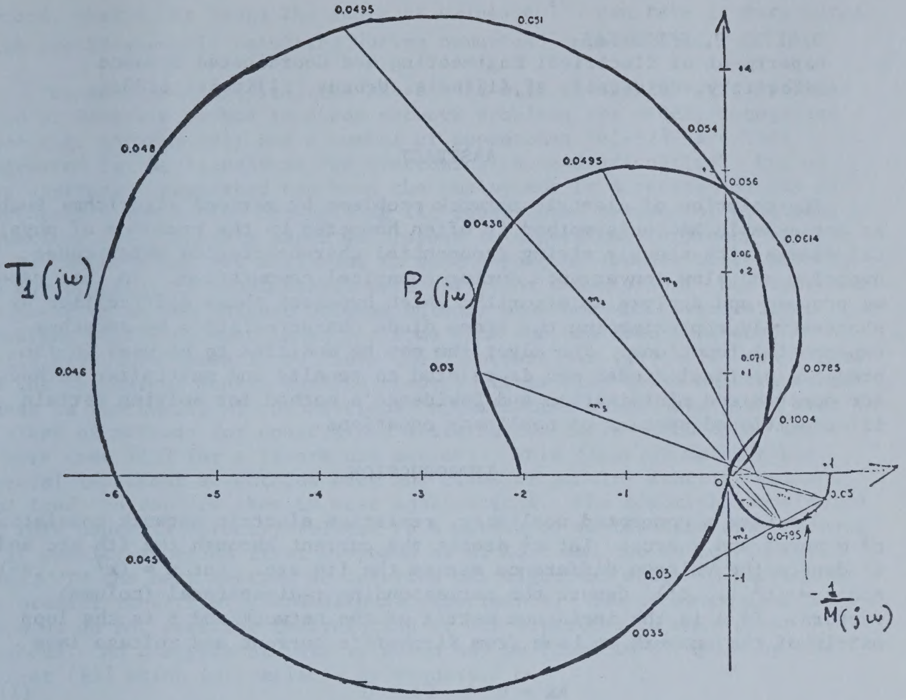


Figure 9: Nyquist plots of T_1,P_2 and $-\frac{1}{M}$ for the system represented in Figure 5.

A NEW ALGORITHM FOR ANALYSIS OF NONLINEAR RESISTIVE NETWORKS*

DIMITRI P. BERTSEKAS
Department of Electrical Engineering and Coordinated Science
Laboratory, University of Illinois, Urbana, Illinois 61801

ABSTRACT

The solution of electric network problems by various algorithms such as for example Newton's method is often hampered by the presence of physical diodes with steeply rising exponential characteristics which cause overflow and slow convergence during numerical computation. In this paper we propose and analyze an algorithm which bypasses these difficulties by successively approximating the steep diode characteristics by smoother exponential functions. The algorithm may be modified to be used in the presence of ideal diodes and is related to penalty and multiplier methods for constrained minimization and Davidenko's method for solving certain ill-conditioned systems of nonlinear equations.

1. INTRODUCTION

Consider a connected nonlinear, resistive electric network consisting of m nodes and r arcs. Let x^i denote the current through the ith arc and t^i denote the voltage difference across the ith arc. Let $x = (x^1,\dots,x^r)'$ and $t = (t^1,\dots,t^r)'$ denote the corresponding r-dimensional (column) vectors. If A is the incidence matrix of the network and B is the loop matrix of the network we have from Kirchoff's current and voltage laws

$$Ax = 0 \quad , \quad Bt = 0 \tag{1}$$

As is well known (see e.g. [1]), the equations above yield a set of r independent equations in (x,t). Suppose that s, with $1 \leq s < r$, out of the r arcs of the network consist of physical diodes with exponential characteristics of the form

$$g_i(x^i,t^i) = x^i - p_i(e^{q_i t^i} - 1) = 0 \quad i = 1,\dots,s \tag{2}$$

where p_i and q_i are given positive scalars while there are (r-s) additional equations relating x and t of the form

$$f_i(x,t) = 0 \quad i = s+1,\dots,r \tag{3}$$

where f_i are continuously differentiable functions.

The numerical solution of the system of equations (1), (2), (3) can be attempted by application of Newton's method. However, this approach in practice often meets with formidable difficulties when the coefficients q_i in the diode characteristic equations (2) are large (a typical range of values for transistor circuits is $q_i = 30 - 40$). First, as predicted by Kantorovich's theorem (see e.g. [2]), the region around a solution where Newton's method exhibits fast convergence depends strongly on the magnitude of the second derivatives of the functions g_i and f_i. When q_i is large the second derivative of g_i with respect to t_i is very large even for moderate positive values of t^i. As a result Newton's method can converge very slowly unless the starting point is very near a solution.

Second, when q_i is large the range of values $e^{q_i t^i}$ can take is very large with overflow easily resulting during numerical computation.

The numerical difficulties associated with straightforward application of Newton's method in diode network problems are widely recognized (see e.g. Calahan [3]) and a number of approaches [4]-[7] have been suggested in the literature for overcoming these difficulties. One of the approaches suggested has been the employment in a particular way of Davidenko's method [8] for solving ill-conditioned systems of nonlinear equations [9]. In this paper we propose an algorithmic approach similar to Davidenko's but yet different in some important aspects which result, in our opinion, in considerably more efficient computation. Our approach is based on an equivalence between network problems and optimization problems which has been known for some time and has been sharpened to a definitive form recently by Minty [10], Rockafellar [11], [12] and others in the context of modern duality theory. It is also based on some recent ideas in the theory of computational optimization which have given rise to a class of methods for constrained optimization called methods of multipliers (see [13] for a literature survey). This class of methods has several important advantages over the class of penalty function methods and tends to replace them in most applications. The algorithm considered in this paper is directly related to a method of multipliers with exponential penalty function first proposed by B. Kort and the author in 1972 [14]. In the next section we describe our algorithm while in Section 3 we present results of computational experiments. Due to space limitation we do not describe the relation of our algorithm to multiplier methods. An analysis of this relation may be found in a report by the author [15] which is available on request.

2. THE ALGORITHM

Consider the possibility of replacing equations (2) by equations of the form

$$\tilde{g}_i(x^i,t^i,y^i,\varepsilon) = 0 \quad i = 1,\ldots,s$$

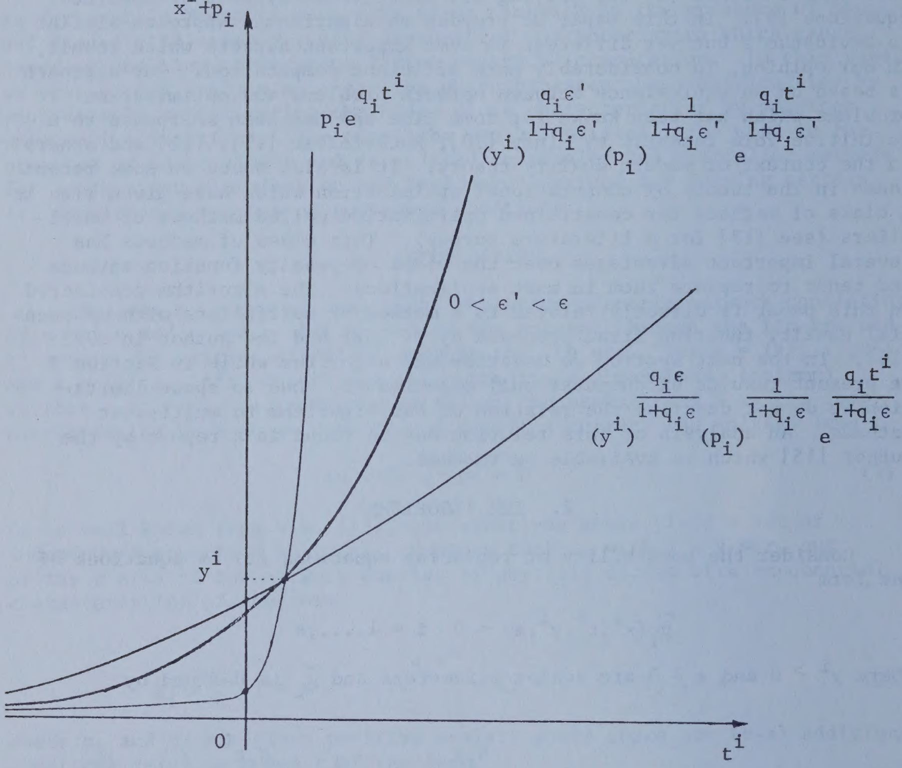
where $y^i > 0$ and $\varepsilon \geq 0$ are scalar parameters and $\tilde{g}_i$ is defined by

$$\tilde{g}_i(x^i,t^i,y^i,\varepsilon) = x^i - (y_i)^{\frac{q_i \varepsilon}{1+q_i \varepsilon}} (p_i)^{\frac{1}{1+q_i \varepsilon}} e^{\frac{q_i t^i}{1+q_i \varepsilon}} + p_i . \tag{5}$$

The functions g_i and $\tilde{g}_i$ are depicted in Figure 1. Notice that

$$\lim_{\varepsilon \to 0^+} \tilde{g}_i(x^i,t^i,y^i,\varepsilon) = g_i(x^i,t^i) \ \forall\, x^i,t^i \in R\, ,\ y^i > 0,\ i = 1,\ldots,s \tag{6}$$

In fact it is easy to see that if $\{x_k^i\}$, $\{t_k^i\}$ are convergent sequences with $x_k^i \to \bar{x}^i$, $t_k^i \to \bar{t}^i$, $\{y_k^i\}$ is any sequence of positive numbers bounded



$x^i + p_i$
$p_i e^{q_i t^i}$
$(y^i)^{\frac{q_i \epsilon'}{1+q_i \epsilon'}} (p_i)^{\frac{1}{1+q_i \epsilon'}} e^{\frac{q_i t^i}{1+q_i \epsilon'}}$
$0 < \epsilon' < \epsilon$
$(y^i)^{\frac{q_i \epsilon}{1+q_i \epsilon}} (p_i)^{\frac{1}{1+q_i \epsilon}} e^{\frac{q_i t^i}{1+q_i \epsilon}}$
y^i
0
t^i

Figure 1

above and bounded away from zero, $\{\epsilon_k\}$ is a sequence of non-negative numbers and furthermore

$$\epsilon_k \to 0, \ \tilde{g}_i(x_k^i, t_k^i, y_k^i, \epsilon_k) \to 0$$

then we must have

$$g_i(\bar{x}^i, \bar{t}^i) = 0.$$

Based on the observations above, a possible approach for solving the system equations (1)-(3) is to specify sequences $\{y_k^i\}$, $i = 1,\ldots,s$, $\{\epsilon_k\}$ and $\{\gamma_k\}$ with

$$0 < \delta \le y_k^i \le \Delta\,, \quad i = 1,\ldots,s \quad \text{for some } \delta,\Delta \tag{7}$$

$$0 \le \epsilon_{k+1} \le \epsilon_k \quad , \quad \forall\, k \quad , \quad \epsilon_k \to 0 \tag{8}$$

$$0 \le \gamma_{k+1} \le \gamma_k \quad , \quad \forall\, k \quad , \quad \gamma_k \to 0 \tag{9}$$

and find for each index k vectors (x_k, t_k) solving approximately the system of equations (1), (3), (4) in the sense that

$$|\tilde{g}_i(x_k^i, t_k^i, y_k^i, \epsilon_k)| \le \gamma_k \qquad i = 1,\ldots,s \tag{10}$$

$$|f_i(x_k, t_k)| \le \gamma_k \qquad i = s+1,\ldots,r \tag{11}$$

$$\|Ax_k\| \le \gamma_k, \quad \|Bt_k\| \le \gamma_k \tag{12}$$

where $\|\cdot\|$ denotes the usual Euclidean norm. Then any limit points $\bar{x}$, $\bar{t}$ of the sequences $\{x_k\}$, $\{t_k\}$ constitute a solution of the network equations (1)-(3). In other words successive approximate solution (within γ_k) of the system of equations (1), (3), (4) yields in the limit solutions of the system of equations (1)-(3).

Notice that finding (x_k, t_k) satisfying (10)-(12) by means of Newton's method will not present difficulties for moderately large values of ϵ_k since now the exponential in equation (5) rises much less steeply than in the diode equation (2). Thus by starting with relatively large values of ϵ (say $\epsilon = 1$) and progressively reducing ϵ towards zero one gradually approaches a solution of the problem in a manner which alleviates the effects of "ill-conditioning" much in the spirit of Davidenko's method and penalty function methods in constrained minimization.

An important additional feature of the method proposed is the introduction of the scalar parameters y_k^i, $i = 1,\ldots,s$. While the algorithm would work if the scalars y_k^i were held constant (say fixed at the value $y_k^i = 1$

for all k) we propose to update y^i_k according to the equation

$$y^i_{k+1} = (y^i_k)^{\frac{q_i \epsilon_k}{1+q_i \epsilon_k}} (p_i)^{\frac{1}{1+q_i \epsilon_k}} e^{\frac{q_i}{1+q_i \epsilon_k} t^i_k} \qquad i = 1,\ldots,s \tag{13}$$

Notice that from (5), (10) and (13) we have for all k

$$|x^i_k + p_i - y^i_{k+1}| \leq \gamma_k \qquad i = 1,\ldots,s \tag{14}$$

As a result if the sequence $\{t^i_k\}$ generated by the algorithm is bounded we have (using (5), (10)) that the sequence $\{x^i_k + p_i\}$ after some sufficiently large index is positive and bounded away from zero. It follows, in view of (14) and the fact $\gamma_k \to 0$, that the sequence $\{y^i_{k+1}\}$ is bounded away from zero for every i = 1,...,s as required by (7). It turns out that in many cases the algorithm proposed is much more efficient when the updating formula (13) is utilized than when the parameters y^i_k are left constant. In fact the algorithm may work satisfactorily even if the parameter ϵ_k is not decreased to zero but rather is held constant at some positive value. In other words convergence may be induced by iteration (13) alone. This fact has been proved for the case of monotone networks (see [15]). For other cases there are no analytical results presently available. However, experimental evidence supports the conjecture that the employment of (13) will accelerate convergence in the great majority of cases of interest.

It is worth noting that, with minor modifications, the algorithm can be also applied to problems with ideal rather than physical diodes, i.e. for diodes with characteristic equation

$$x^i = \begin{cases} 0 & \text{if } t^i \leq 0 \\ +\infty & \text{if } t^i > 0 \end{cases}$$

The approximating function $\tilde{g}_i$ in this case takes the form

$$\tilde{g}_i(x^i, t^i, y^i, \epsilon) = x^i - y^i e^{\frac{t^i}{\epsilon}} = 0$$

and the multiplier iteration (13) takes the form

$$y^i_{k+1} = y^i_k \, e^{\frac{t^i_k}{\epsilon_k}} \qquad i = 1,\ldots,s$$

We end this section by demonstrating pictorially how the algorithm will work for the case of a simple example network.

Example: Consider the network shown in Figure 2 where the diode shown is either ideal or has an exponential characteristic. We assume that the solution of the approximating network is exact i.e. $\gamma_k \equiv 0$.

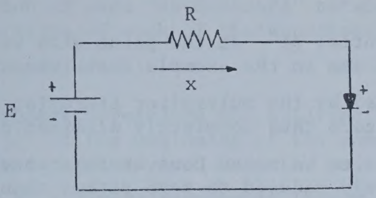


Figure 2

Let x and t denote the current and voltage difference across the diode and let them be related by

$$x = p(e^{qt} - 1)$$

The successive iterations of our algorithm (with $\gamma_k \equiv 0$) are shown in Figure 3

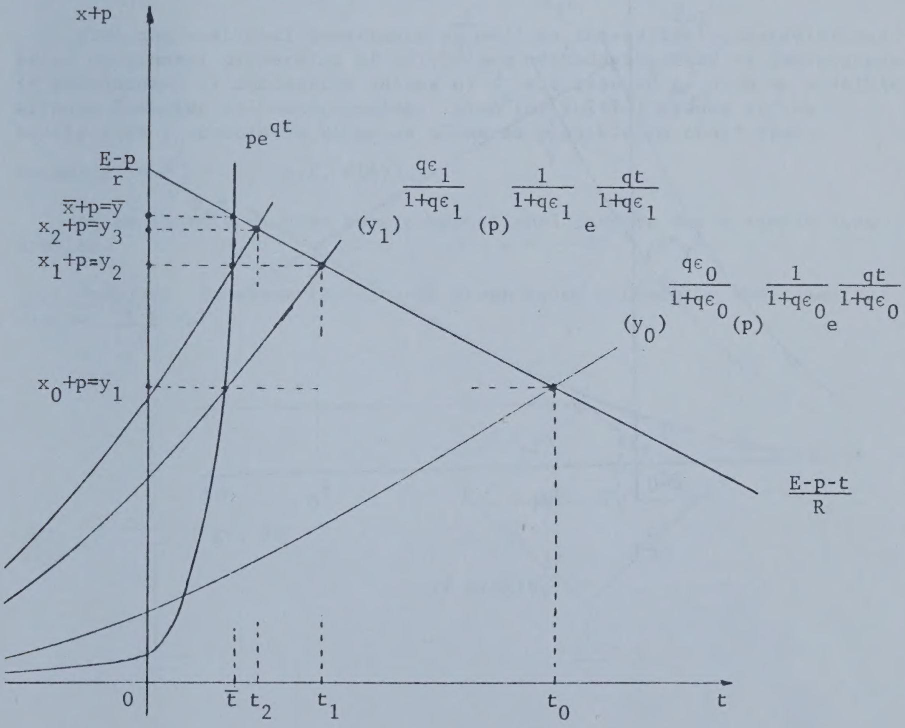


Figure 3

Notice the mechanism by which the multipliers y_k are generated and the fact that the "approximating" exponential characteristic

$$(y_k)^{\frac{q\epsilon_k}{1+q\epsilon_k}}(p)^{\frac{1}{1+q\epsilon_k}}e^{\frac{qt}{1+q\epsilon_k}}$$

intersects the diode exponential pe^{qt} at the point with vertical coordinate $y_k = x_{k-1}+p$. It is easy to see in the example above that convergence to the pair $(\bar{x},\bar{t})$ can be induced by the multiplier iteration (13) alone without the need to decrease ϵ_k to zero thus completely eliminating the ill-conditioning problem. It is to be noted however that convergence will be faster when ϵ_k is successively reduced to zero rather than kept constant.

Consider now the case where the diode in Figure 2 is an ideal diode. Then the solution of the problem is trivially $\bar{x} = \frac{E}{R}$, $\bar{t} = 0$. Figure 4 shows how our algorithm will find this solution.

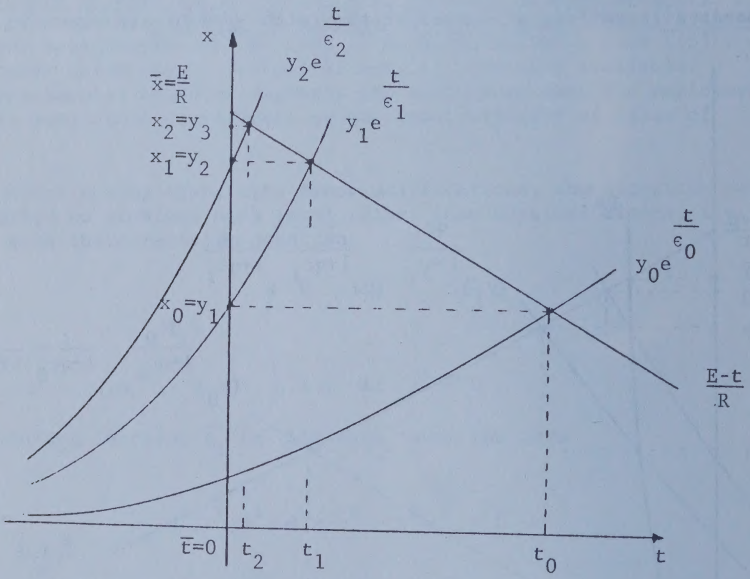

x
$\bar{x}=\frac{E}{R}$
$x_2=y_3$
$x_1=y_2$
$x_0=y_1$
$y_2e^{\frac{t}{\epsilon_2}}$
$y_1e^{\frac{t}{\epsilon_1}}$
$y_0e^{\frac{t}{\epsilon_0}}$
$\frac{E-t}{R}$
$\bar{t}=0$
t_2
t_1
t_0
t

Figure 4.

3. COMPUTATIONAL ASPECTS AND RESULTS

The algorithm presented in this paper has been utilized to solve a few test problems. Generally the method performed in a reliable manner. In all problems the pure form of Newton's method for solving nonlinear equations (i.e. with stepsize equal to one) was utilized throughout. The algorithmic solution of each "approximate" network problem was terminated when the absolute value of each of the equations was less than 10^{-5} ($\gamma_k \equiv 10^{-5}$ in (10)-(12)).

A special automatic procedure was utilized to determine the sequence of parameters $\{\epsilon_k\}$. At the beginning of the computation a sufficiently large value ϵ_0 was determined such that the values of all equations at the initial point (x_0, t_0, y_0) were reasonable (say $< 10^3$). At the beginning of the kth solution cycle a trial value $\epsilon_k = \frac{\epsilon_{k-1}}{8}$ was adopted and this value was decreased or increased depending on whether the values of all equations at the point (x_{k-1}, t_{k-1}, y_k) was within certain ranges (in our problems we decreased ϵ by a factor of 8 if all initial equation values were less than 10^{-3} and we increased ϵ by a factor of 2 if some initial equation values were greater than unity. We took however always $\epsilon_k \leq 0.5\ \epsilon_{k-1}$ so that the sequence $\{\epsilon_k\}$ was always monotonically decreasing). This automatic procedure for adjusting the approximation parameter ϵ is aimed at striking a reasonable balance between fast convergence of the algorithm and reduction of ill-conditioning.

From computational experience as well as theoretical considerations based on general properties of multiplier methods the rate of convergence is accelerated if successive values of ϵ_k are reduced as much as possible without inducing ill-conditioning. Also the initial values of the multipliers y_0 should be taken as close as possible to their final values $\bar{y}^i = \bar{x}^i + p_i$ (c.f. (14)).

As an example we give some computational results for a simple test problem.

Test Problem: Consider the network given below containing two identical diodes:

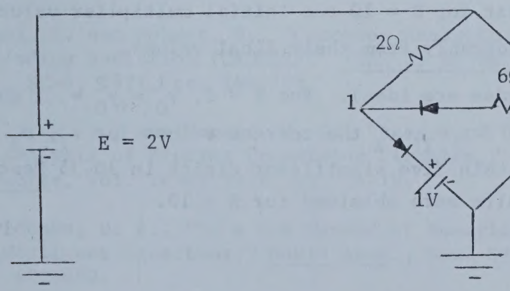


Figure 5.

Case 1: For this case the diodes have identical exponential characteristics of the form $x = .1 \times 10^{-14}(e^{40t} - 1)$. The two equations for the voltages v_1, v_2 at the points 1 and 2 shown in the figure are as follows:

$$E - 2v_2 - p[e^{q(13v_2 - v_1 - 6E)} - 1] = 0 \quad (15)$$

$$1.5E - 2v_2 - 0.5v_1 - p[e^{q(v_1 - 1)} - 1] = 0 \quad (16)$$

where $p = .1 \times 10^{-14}$, $q = 40$, $E = 2$.

When the pure form of Newton's method was utilized to solve the equations above we obtained convergence in 12 iterations from the initial point $v_1 = v_2 = 1$ to the final point $v_1 = 1.8052$, $v_2 = 1.0000$. However for the initial point (3,0) Newton's method was progressing very slowly and failed to converge after 20 iterations while for the initial points (0,4) and (-2,6) overflow occurred immediately.

We now give in Table 1 the results of the computation using our method. The table gives the parameter ϵ_k for the kth network problem, the multiplier values y_k^1, y_k^2 (y_k^1, y_k^2 refer to the diodes in equations (15) and (16) respectively), the values of v_1, v_2 obtained, and the number of iterations of Newton's method. The initial point was $v_1 = v_2 = 1$. Notice that the number of iterations is 8 (four less than for the ordinary Newton's method). For the initial point $v_1 = 3$, $v_2 = 0$ ($y_0^1 = y_0^2 = .1$) the number of iterations was 8. For $v_1 = 0$, $v_2 = 4$ ($y_0^1 = y_0^2 = .1$) the number was 15. For $v_1 = -2$, $v_2 = 6$ ($y_0^1 = y_0^2 = .1$) the number was 11. For $v_1 = 5$, $v_2 = 8$ ($y_0^1 = y_0^2 = .1$) the number was 12. For $v_1 = 10$, $v_2 = 5$ ($y_0^1 = y_0^2 = .1$) the number was 15. Thus there was no difficulty in solving the problem for widely varying initial conditions and furthermore the number of iterations required was not extremely sensitive to the initial values of v_1 and v_2. Similar results were obtained for E = 10 and the same initial conditions although the number of iterations required was higher (around 30). This may be attributed to the fact that for E = 10 the initial multiplier values $y_0^1 = y_0^2 = .1$ differ considerably from their final values.

Case 2: Here the two diodes are ideal. For E = 2, $y_0^1 = y_0^2 = .1$ and a variety of initial values for v_1, v_2 the correct values for v_1, v_2 ($v_1 = v_2 = 1$) was obtained within five significant digits in 30-35 Newton iterations. Similar results were obtained for E = 10.

k	ϵ_k	Number of Iterations	(y_k^1,y_k^2)	(v_1,v_2)
0	1.000	3	(.10 , .10)	(1.8311, .99097)
1	.1250	2	(.0181,.1025)	(1.8016, .99999)
2	.0156	2	(.0000,.0992)	(1.8050,1.0000)
3	.0002	1	(.0000,.0975)	(1,8052,1.0000)

Table 1

4. CONCLUSIONS

The algorithm presented in this paper represents a simple way to overcome the numerical difficulties associated with steeply rising diode characteristics in resistive network problems. As such it can be viewed as a method for increasing the reliability of network analysis programs. Since the nature of the algorithm is such that the basic network structure of the problem is not affected, the procedure proposed can be very easily incorporated within existing programs.

REFERENCES

[1] Balabanian, N. and Bickart, T. A., *Electrical Network Theory*, John Wiley, New York, 1969.

[2] Kantorovich, L. V. and Akilov, G. P., *Functional Analysis in Normed Spaces*, Pergamon Press, London, 1964.

[3] Calahan, D. A., *Computer-Aided Network Design*, McGraw-Hill, Inc., New York, 1972.

[4] Branin, F. H., Hogsett, G. R., Lunde, R. L. and Kugel, L. E., "ECAP II-A New Electronic Circuit Analysis Program," *IEEE J. on Solid State Circuits*, Vol. SC-6, 1971, pp. 146-166.

[5] Brown, G. C., "DC Analysis of Nonlinear Networks," *Electron. Lett.*, Vol. 5, 1969, pp. 374-375.

[6] Nagel, L. and Rohrer, R., "Computer Analysis of Nonlinear Circuits, Excluding Radiation (CANCER)," *IEEE J. of Solid State Circuits*, Vol. SC-6, 1971, pp. 166-182.

[7] Calla, W. J., and Howard, W. G., "Bias-3, A Program for the Nonlinear DC Analysis of Bipolar Transistor Circuits," *IEEE J. of Solid State Circuits*, Vol. SC-6, 1971, pp. 14-19.

[8] Davidenko, D. F., "On a New Method of Numerical Solution of Systems of Nonlinear Equations," *Dokl. Akad.*, Nauk SSSR, Vol. 88, 1953, pp. 601-602.

[9] Cermak, I. A., "DC Solution of Nonlinear State Space Equations in Circuit Analysis," IEEE Trans. on Circuit Theory, Vol. CT-18, 1971, pp. 312-314.

[10] Minty, G. J., "Monotone Networks," Proc. Royal Soc. London, Series A 257, 1960, pp. 194-212.

[11] Rockafellar, R. T., "Convex Programming and Systems of Elementary Monotonic Relations," J. of Math. Anal. & Appl., Vol. 19, 1967, pp. 543-564.

[12] Rockafellar, R. T., Optimization in Networks, Lecture Notes, University of Grenoble, France, 1974.

[13] Bertsekas, D. P., "Multiplier Methods: A Survey," Automatica, March 1976, to appear.

[14] Kort, B. W. and Bertsekas, D. P., "A New Penalty Function Method for Constrained Minimization," Proc. 1972 IEEE Conf. on Decision and Control, New Orleans, pp. 162-166.

[15] Bertsekas, D. P., "A New Algorithm for Solution of Resistive Networks Involving Diodes," Coordinated Science Laboratory Working Paper, University of Illinois, Urbana, 1975.

---

[*] Work supported in part by the Joint Services Electronics Program (U. S. Army, U. S. Navy, and U. S. Air Force) under Contract DAAB-07-72-C-0259, and in part by the National Science Foundation under Grant ENG 74-19332.

TOLOPT - A PROGRAM FOR OPTIMAL, CONTINUOUS OR DISCRETE, DESIGN CENTERING AND TOLERANCING

J.W. BANDLER
Group on Simulation, Optimization and Control
McMaster University, Hamilton, Canada

J.H.K. CHEN
Bell-Northern Research
Ottawa, Canada

P. DALSGAARD
Institute of Electronic Systems
Aalborg University, Aalborg, Denmark

P.C. LIU
Bell-Northern Research
Verdun, P.Q., Canada

ABSTRACT

This paper presents a user-oriented computer program package called TOLOPT (TOLerance OPTimization), which can solve continuous and/or discrete worst-case tolerance assignment problems. Worst-case vertices can be automatically selected and optimization will lead to the most favorable nominal design simultaneously with the largest possible tolerances on specified toleranced components. The program, which is available, contains recent techniques and algorithms for nonlinear programming.

INTRODUCTION

TOLOPT is a package of subroutines which can solve continuous and discrete worst-case tolerance assignment problems simultaneously with the selection of the most favorable nominal design [1-3]. The package is designed to handle the objective function, performance specifications, and parameter constraints in a unified manner such that any of the nominal values or tolerances (relative or absolute) can be fixed or varied automatically at the user's discretion. Time-saving techniques for choosing constraints (vertices selection) are incorporated. The routine involved also checks assumptions and performs worst-case analyses.

The continuous and (optional) discrete optimization methods are programmed in such a way that they may be used as a separate unit. This part, called DISOP2 and incorporating several optional features, is an updated version of DISOPT, which has been successfully applied in many different areas [3 - 6]. Dakin's tree search for discrete problems [7], efficient gradient minimization of functions of many variables by a recent quasi-Newton method [8] and recent developments in least pth approximation by Bandler and Charalambous [9 - 12] are employed. Extrapolation is also featured [13]. The Fortran IV package typically requires 64000 octal words on the CDC 6400.

FEATURES OF TOLOPT

TOLOPT organizes input data and coordinates other subprograms such

---

This work was supported by the National Research Council of Canada under Grant A7239 and by the Danish Council for Scientific and Industrial Research through support to P. Dalsgaard.

as DISOP2, VERTST, CONSTR, USERCN, COSTFN and NETWRK. Subroutine DISOP2 is a general program for continuous and discrete non-linear programming problems. Subroutine VERTST eliminates the inactive vertices of the tolerance region. Subroutine CONSTR sets up the constraint functions based on the response specifications, component bounds and other constraints supplied in the user subroutine USERCN. Subroutine COSTFN computes the cost function. The user has the option of supplying his own subroutine to define other cost functions. The user supplied subroutine NETWRK returns the network responses and the partial derivatives. In the user supplied subroutine USERCN the user has to define whatever extra constraints he needs and the corresponding partial derivatives. It should be noted that the constraints given in USERCN are not checked against the worst-case vertices.

Table I is a summary of the features and options currently incorporated in TOLOPT.

The objective function we have investigated and implemented [1 - 3] is the weighted summation of the inverses of the relative or the absolute tolerances. The weighting factors may (as default values) be taken as one, but the user can specify his own set of weighting factors.

Various schemes have been developed to identify or to predict the most critical vertices that are likely to give rise to active constraints. Our proposed schemes will eliminate all but one vertex for each constraint function in the most favourable conditions. When monotonicity assumptions [2, 14] are not sufficient to describe the function behaviour, our scheme will increase the number of vertices until, at worst, all vertices are included.

Two major schemes of increasing complexity are programmed in the subroutine VERTST [3]. One involves vertices $\underset{\sim}{\phi}^a = \underset{\sim}{\phi}^0 - \varepsilon_j \underset{\sim}{u}_j$ and $\underset{\sim}{\phi}^b = \underset{\sim}{\phi}^0 + \varepsilon_j \underset{\sim}{u}_j$. Here, $\underset{\sim}{\phi}^0$ is the nominal point and ε_j the tolerance on the jth component. $\underset{\sim}{u}_j$ is the jth unit vector and $j \in I_\phi$, where $I_\phi \overset{\Delta}{=} \{1,2,...,k\}$ is the index set for the network components. Another involves all vertices. Also, the special case which occurs for $\underset{\sim}{\phi}^a = \underset{\sim}{\phi}^b$, has been programmed. In this case only one vertex is considered for each sample point.

The user decides on which vertices selection scheme he wants to use as well as the maximum number of allowable calls for the scheme selected for the updating procedure. He may, if he wishes, bypass the whole subroutine by supplying his own vertices or set up his own strategy of vertices selection. Furthermore, the user decides on the maximum number of vertices allowable at each sample point. If more than the maximum allowable numbers are detected, the subroutine selects the ones corresponding to the lowest constraint value arranged in ascending order.

After printing out the detected vertices and the value of the corresponding constraints, the user has the possibility of eliminating further vertices by considering the relative magnitude of the constraints.

As an option the TOLOPT program can be used for vertices detection only. The program will print out the detected vertices and the value of the corresponding constraints such that the user has the possibility manually to eliminate vertices using his own judgement. The user has the possibility of supplying his own set of active vertices in two different ways.

Before using the automated vertex selection an initial feasibility check is performed to check the feasibility of the nominal design. The outcome from this feasibility check is used as a starting point in the tolerance assignment problem. If a feasible nominal point is not attainable, the user has to relax some specifications or change his design.

The different optimization methods incorporated [9 - 13] employ the least pth approach. Once the constraints have been selected, optimization is started with a small value of p and α (a parameter associated with the minimax approach to nonlinear programming [11]). The routine for updating constraints is called whenever the α value is updated and/or each time new constraints have been added. We add new values of $\underset{\sim}{\mu}$ (a vector identifying a vertex) to existing ones without any eliminations, for stability. When the maximum number of calls is exceeded or when there is no change of values for consecutive calls the program goes to the final optimization with the set of vertices chosen.

Using all the detected vertices could, depending on the problem under investigation, easily involve so many constraints that the optimization would be very time consuming. This could, however, for some problems, be overcome by specifying a sufficiently large but reasonable limit. In such cases the updating and optimization procedure will converge if the vertices, which are active at the solution, are not discarded during updating. The same convergence should occur if manual elimination by the user is performed without discarding vertices which are active at the solution.

It should be pointed out that vertices which are detected at an early stage of the updating procedure need not be active at the solution and vice-versa. The final solution is worst-case only at the chosen sample points.

The solution process may provide valuable information to the user, e.g., parameter or frequency symmetry, which could be useful in order to reduce the number of active vertices.

EXAMPLE

We consider a simple voltage divider [4, 15] with resistances of ϕ_1 and ϕ_2, a transfer function of $\phi_2/(\phi_1+\phi_2)$ and input resistance $\phi_1+\phi_2$. The design specifications are $0.46 \leq \phi_2/(\phi_1+\phi_2) \leq 0.53$ and $1.85 \leq \phi_1+\phi_2 \leq 2.15$. In the case of the discrete problem the set of obtainable discrete values for the tolerances of ϕ_1 and ϕ_2 are 1,3,5,10,15 percent.

A typical main program to supply the values and proper dimensioning for the parameters in the argument list of subroutine TOLOPT and the common statements /TOL/ and /DEFAULT/ is displayed in Fig. 1. Fig. 2 shows the subroutine NETWRK and Fig. 3 illustrates USERCN for a constraint inactive at the solution. Typical printouts of data and the gradient check are shown in Figs. 4 and 5, respectively. Results of continuous and discrete optimizations are shown in Fig. 6.

In this example all four known vertices are supplied and TOLOPT goes directly to the final optimization.

CONCLUSIONS

We have presented an efficient user-oriented program for worst-case tolerance optimization, particularly suited to circuit design. It is based on work carried out by Chen [5], Liu [2] and Bandler, Liu and Chen [3]. The package has been under continuous development to make it sufficiently user-oriented. This has been to some extent at the expense of the greater efficiency which can be realized by a more specialized program. Running times of the package can vary significantly according to the various termination and error criteria used as data. This is particularly true in the generation of the tree structure in a discrete optimization and the interpretation of the solutions as being feasible, discrete, etc.

A detailed report with a complete documented listing is available from the first author at a nominal charge [16].

REFERENCES

[1] J.W. Bandler, "Optimization of design tolerances using nonlinear programming", J. Optimization Theory and Applications, vol. 14, 1974, pp. 99-114.

[2] P.C. Liu, "A theory of optimal worst-case design embodying centering, tolerancing and tuning, with circuit applications", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-87, May 1975.

[3] J.W. Bandler, P.C. Liu and J.H.K. Chen, "Worst case network tolerance optimization", IEEE Trans. Microwave Theory Tech., vol. MTT-23, Aug. 1975, pp. 630-641.

[4] J.W. Bandler and J.H.K. Chen, "DISOPT - a general program for continuous and discrete nonlinear programming problems", Int. J. Systems Science, vol. 6, 1975, pp. 665-680.

[5] J.H.K. Chen, "DISOPT- a general program for continuous and discrete nonlinear programming problems", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-29, March 1974 (Revised June 1975).

[6] J.W. Bandler, B.L. Bardakjian and J.H.K. Chen, "Design of recursive digital filters with optimized word length coefficients", Computer Aided Design, vol. 7, July 1975, pp. 151-156.

[7] R.J. Dakin, "A tree-search algorithm for mixed integer programming problems", Computer J., vol. 8, 1966, pp. 250-255.

[8] R. Fletcher, "FORTRAN subroutines for minimization by quasi-Newton methods", Atomic Energy Research Establishment, Harwell, Berkshire, England, Report AERE-R7125, 1972.

[9] J.W. Bandler and C. Charalambous, "Practical least pth optimization of networks", IEEE Trans. Microwave Theory Tech., vol. MTT-20, Dec. 1972, pp. 834-840.

[10] C. Charalambous and J.W. Bandler, "New algorithms for network optimization", IEEE Trans. Microwave Theory Tech., vol. MTT-21, Dec. 1973, pp. 815-818.

[11] J.W. Bandler and C. Charalambous, "Nonlinear programming using minimax techniques", J. Optimization Theory and Applications, vol. 13, June 1974, pp. 607-619.

[12] C. Charalambous, "A unified review of optimization", IEEE Trans. Microwave Theory Tech., vol. MTT-22, March 1974, pp. 289-300.

[13] W.Y. Chu, "Extrapolation in least pth approximation and nonlinear programming", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-71, Dec. 1974.

[14] J.W. Bandler and P.C. Liu, "Some implications of biquadratic functions in the tolerance problem", IEEE Trans. Circuits and Systems, vol. CAS-22, May 1975, pp. 385-390.

[15] B.J. Karafin, "The optimum assignment of component tolerances for electrical networks", B.S.T.J., vol. 50, April 1971, pp. 1225-1242.

[16] J.W. Bandler, J.H.K. Chen, P. Dalsgaard and P.C. Liu, "TOLOPT- a program for optimal, continuous or discrete, design centering and tolerancing", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-105, Sept. 1975.

TABLE I

SUMMARY OF FEATURES AND OPTIONS

Features	Type	Options
Design parameters	Tolerance and nominal	Variable or fixed. Relative or absolute tolerances
Objective function	Cost	Reciprocal of relative and/or absolute tolerances Other
Vertices selection*	Gradient direction strategy	
Constraints	Specifications on functions of network parameters	Upper and/or lower
	Network parameter bounds	
	Other constraints	As many as required
Nonlinear programming	Bandler-Charalambous minimax	Least pth optimization algorithms
	Exterior-point	
Solution feasibility check*	Least pth	Discrete problem. Continuous and discrete problem
Unconstrained minimization method	Quasi-Newton	Gradient checking at starting point by numerical perturbation
Discrete optimization*	Dakin tree-search	Reduction of dimensionality. User supplied or program determined initial upper bound on objective function. Single or multiple optimum discrete solution. Uniform or nonuniform quantization step sizes

* These features are optional and may be bypassed.

```
      PROGRAM TESTVOL(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT)
      DIMENSION X(4), EPS(4), G(4), PS(1), XB(4), IX(4), X1(4), X2(4), W(16),
     *    H(10), XE(4,1,1), INDX(4), GF(4)
      DIMENSION IAA(50), IBB(50), A(50), T1(50), T1P(50)
      DIMENSION NSTEP(2), QSTEP(2), DISCR(2,5), XU(2), XL(2), ID(25), IB(2,25)
     *   ,ICHECK(25), IVAR(25), P1(25), P2(25), ESTD(25), AL(25)
      DIMENSION Z(4), I1(4), I2(4), AZ(2), AX(2), MU(2,10), NV(10), SAMPT(3,10)
     *   ,GRAD(2), PL(2), PU(2), W1(4), CW(2), IB1(2), SG(10)
      DIMENSION GPHI(4,24), PHI(24), I3(10), I4(10)
      COMMON /TOL/ IUPD, ISCEME, IWORST, IPRINT, IDATA, IOPT1, IOPT2, IOPT3,
     *   IOPT4, IOPT5, IOPT6, IOPT7, ND2, ND3, ND4, ND5, MAX, MAXNOD, ICON, NDIM,
     *   NSP, MAXVN, NVSUM, NEC, ND1, ND6
      COMMON /DEFAULT/ EST, EST1, AO, AI, XMAL, ZERO, ETA, INSOLN, BSOLN
      .
      .
      .
      KP=2*KT
      READ 3, ((SAMPT(I,J), I=1,3), J=1,NVSUM)
    3 FORMAT(3F5.2)
      READ 4, ((MU(I,J), J=1,NVSUM), I=1, KT)
      READ 4, (NV(I), I=1,NVSUM)
    4 FORMAT(4I3)
      CALL TOLOPT (NR,KT,KR,KD,KP,NP,Z,I1,I2,AZ,AX,MU,NV,SAMPT,
     *   GRAD,PL,PU,W1,CW,IB1,SG,I3,I4,X,EPS,G,PS,XB,IX,X1,X2,
     *   W,H,XE,INDX,GF,IAA,IBB,A,T1,T1P,NSTEP,QSTEP,DISCR,XU,XL,ID,
     *   IB,ICHECK,IVAR,P1,P2,ESTD,AL,GPHI,PHI)
      STOP
      END
```

Fig. 1. Main program for the example.

```
      SUBROUTINE NETWRK (AX,OM,RSP,GR,IG)
      DIMENSION AX(1), GR(1)
      A=AX(1) + AX(2)
      A2=A**2
      T=AX(2)/A
      KV=IFIX(OM)
      GO TO (1,2,2,1), KV
    1 RSP=A
      IF(IG.EQ.0) RETURN
      GR(1)=1.
      GR(2)=1.
      RETURN
    2 RSP=T
      IF(IG.EQ.0) RETURN
      GR(1)=-AX(2)/A2
      GR(2)=AX(1)/A2
      RETURN
      END
```

Fig. 2. Subroutine NETWRK .

```
      SUBROUTINE USERCN (Z,G,GG,NR,KP)
      DIMENSION Z(1), G(1), GG(KP,1)
      G(1)=Z(3)+Z(4)
      GG(1,1)=0.
      GG(2,1)=0.
      GG(3,1)=1.
      GG(4,1)=1.
      RETURN
      END
```

Fig. 3. Subroutine USERCN.

```
INPUT DATA
DISCRETE VALUES FOR THE VARIABLES
Z( 1)  .10000000E+01  .30000000E+01  .50000000E+01  .10000000E+02  .15000000E+02
Z( 2)  .10000000E+01  .30000000E+01  .50000000E+01  .10000000E+02  .15000000E+02
USER SUPPLIED COMPONENTS......TOLERANCE, DISCRETE..........Z( 1)=  .10000000E-01
                              TOLERANCE, DISCRETE..........Z( 2)=  .10000000E-01
                              NOMINAL, CONTINUOUS..........Z( 3)=  .10000000E+01
                              NOMINAL, CONTINUOUS..........Z( 4)=  .10000000E+01
ERROR TOLERANCE IN CONSTRAINTS.............................ZERO= -.10000000E-05
TEST QUANTITIES TO BE USED IN FLETCHER METHOD...........EPS( 1)=  .10000000E-05
                                                        EPS( 2)=  .10000000E-05
                                                        EPS( 3)=  .10000000E-05
                                                        EPS( 4)=  .10000000E-05
ESTIMATE OF LOWER BOUND ON ARTIFICIAL OBJECTIVE FUNCTION.....EST= 0.
INITIAL VALUE OF THE PARAMETER ALPHA.........................AO=  .10000000E+03
MAXIMUM ALLOWABLE VALUE OF THE PARAMETER ALPHA.............XMAL=  .10000000E+08
MULTIPLYING FACTOR IN ALPHA VALUE............................AI=  .10000000E+02
TEST QUANTITY TO BE USED IN NLP ALGORITHM 2/4/5.............ETA=  .10000000E-02
NUMBER OF P VALUES...........................................NP=   1
VALUE(S) OF P USED IN NLP ALGORITHM.....................PS( 1)=  .60000000E+01
FOLLOWING OPTIONS USED
NLP ALGORITHM 4 EMPLOYED-SEQUENCE OF LEAST PTH OPTIMIZATION WITH FINITE VALUES
OF P
(N-1) VARIABLE OPTIMIZATION PERFORMED IN DISCRETE PROBLEM
VERTICES CHECKED ABOUT CONTINUOUS SOLUTION TO OBTAIN AN INITIAL UPPER BOUND
IN DISCRETE PROBLEM
FEASIBILITY CHECKED IN FINAL OPTIMIZATION
PARTITIONING STARTS ON LAST DISCRETE VARIABLE
DATA GIVEN FOR SPECIFIC PROBLEM

     IF       SAMPT(1,IF)     SAMPT(2,IF)      SAMPT(3,IF)
     1       .101000E+01     .185000E+01     -.100000E+01
     2       .201000E+01     .460000E+00     -.100000E+01
     3       .301000E+01     .530000E+00      .100000E+01
     4       .401000E+01     .215000E+01      .100000E+01
```

Fig. 4. Printout of data.

```
GRADIENT CHECK AT NOMINAL STARTING POINT

THE GRADIENTS FROM THE USER SUPPLIED NETWRK HAVE BEEN CHECKED AT THE FIRST
SAMPLE POINT

    ANALYTICAL GRADIENTS     NUMERICAL GRADIENTS     PERCENTAGE ERRORS
         .100000E+01             .100000E+01             .378577E-08
         .100000E+01             .100000E+01             .378577E-08

THE GRADIENTS FROM NETWRK HAVE BEEN CHECKED AT ALL SAMPLE POINTS
THE LARGEST OVERALL DETECTED ERRORS ARE AS FOLLOWS

    ANALYTICAL GRADIENTS     NUMERICAL GRADIENTS     PERCENTAGE ERRORS   SAMPLE POINT
        -.250000E+00            -.250000E+00             .244904E-06          2
         .250000E+00             .250000E+00             .244904E-06          2

THE GRADIENTS FROM THE USER SUPPLIED USERCN HAVE BEEN CHECKED
FOR EACH GIVEN EXTRA CONSTRAINT THE ERRORS ARE AS FOLLOWS

    ANALYTICAL GRADIENTS     NUMERICAL GRADIENTS     PERCENTAGE ERRORS   CONSTRAINT
         .100000E-13             .100000E-13           0.                     1
         .100000E-13             .100000E-13           0.                     1
         .100000E+01             .100000E+01             .378577E-08          1
         .100000E+01             .100000E+01             .378577E-08          1
```

Fig. 5. Check of user supplied gradients.

```
                    RESULTS OF THE FEASIBILITY CHECK

                             NODE NUMBER = 0

    INEQUALITY CONSTRAINTS                        OCCURRING AT

    G( 1) =  .13000000E+00                        SAMPLE POINT 1
    G( 2) =  .35000000E-01                        SAMPLE POINT 2
    G( 3) =  .25000000E-01                        SAMPLE POINT 3
    G( 4) =  .13000000E+00                        SAMPLE POINT 4
    G( 5) =  .19000000E+00                        LOWER BOUND  1
    G( 6) =  .19000000E+00                        UPPER BOUND  1
    G( 7) =  .19000000E+00                        LOWER BOUND  2
    G( 8) =  .19000000E+00                        UPPER BOUND  2
    G( 9) =  .20000000E+01                        EXTRA CONST  1
          NUMBER OF CONSTRAINTS USED =              9
      NUMBER OF VIOLATED CONSTRAINTS =              0
      NUMBER OF FUNCTION EVALUATIONS =              1

                  FOLLOWING IS RESULT OF OPTIMIZATION

                             NODE NUMBER =          0
 ARTIFICIAL UNCONSTRAINED FUNCTION U =        -.11592424E-01
          ACTUAL OBJECTIVE FUNCTION F =         .28569099E+02
    X( 1) =  .26458592E+00  GU( 1)    =         .15487643E-03
    X( 2) =  .26458592E+00  GU( 2)    =         .16120715E-03
    X( 3) =  .10139413E+01  GU( 3)    =         .10367310E-03
    X( 4) =  .99376532E+00  GU( 4)    =        -.10577793E-03
    INEQUALITY CONSTRAINTS                        OCCURRING AT
    G( 1) =  .17155658E-01                       SAMPLE POINT 1
    G( 2) =  .66321421E-06                       SAMPLE POINT 2
    G( 3) =  .66395748E-06                       SAMPLE POINT 3
    G( 4) =  .17425017E-02                       SAMPLE POINT 4
    G( 5) =  .14295958E+00                       LOWER BOUND  1
    G( 6) =  .11507707E+00                       UPPER BOUND  1
    G( 7) =  .12419608E+00                       LOWER BOUND  2
    G( 8) =  .13666543E+00                       UPPER BOUND  2
    G( 9) =  .20077066E+01                       EXTRA CONST  1
          NUMBER OF CONSTRAINTS USED =             9
      NUMBER OF VIOLATED CONSTRAINTS =             0
      NUMBER OF FUNCTION EVALUATIONS =           192
  FINAL VALUE OF THE PARAMETER ALPHA =          .10000000E+05

                   FOLLOWING IS THE OPTIMUM SOLUTION

    Z( 1) =  .70005708E-01
    Z( 2) =  .70005708E-01
    Z( 3) =  .10139413E+01
    Z( 4) =  .99376532E+00

            EXECUTION TIME IN SECONDS =          4.20300
```

Fig. 6. Results of the continuous optimization.

```
          BEST DISCRETE SOLUTION FOUND SO FAR
                                     F =    .40000000E+02

        X( 1) =  .50000000E-01
        X( 2) =  .50000000E-01
        X( 3) =  .10139413E+01
        X( 4) =  .99376532E+00
        INEQUALITY CONSTRAINTS      OCCURRING AT
        G( 1) =  .57321249E-01      SAMPLE POINT  1
        G( 2) =  .99904561E-02      SAMPLE POINT  2
        G( 3) =  .10014581E-01      SAMPLE POINT  3
        G( 4) =  .41908093E-01      SAMPLE POINT  4
        G( 5) =  .16324419E+00      LOWER BOUND   1
        G( 6) =  .13536168E+00      UPPER BOUND   1
        G( 7) =  .14407706E+00      LOWER BOUND   2
        G( 8) =  .15654641E+00      UPPER BOUND   2
        G( 9) =  .20077066E+01      EXTRA CONST   1
NUMBER OF FUNCTION EVALUATIONS =    198
            RESULTS OF THE FEASIBILITY CHECK
                        NODE NUMBER =      8
        INEQUALITY CONSTRAINTS      OCCURRING AT
        G( 1) =  .94253713E-02      SAMPLE POINT  1
        G( 2) = -.34594404E-02      SAMPLE POINT  2
        G( 3) =  .68230156E-02      SAMPLE POINT  3
        G( 4) =  .26401155E-01      SAMPLE POINT  4
        G( 5) =  .16418646E+00      LOWER BOUND   1
        G( 6) =  .13432023E+00      UPPER BOUND   1
        G( 7) =  .95238913E-01      LOWER BOUND   2
        G( 8) =  .14208092E+00      UPPER BOUND   2
        G( 9) =  .19915121E+01      EXTRA CONST   1
    NUMBER OF CONSTRAINTS USED =     10
NUMBER OF VIOLATED CONSTRAINTS =      2
NUMBER OF FUNCTION EVALUATIONS =     10
     EXECUTION TIME IN SECONDS =      .27500
             OPTIMUM DISCRETE SOLUTION FOUND
                          MINIMUM F =      .40000000E+02

        X( 1) =  .50000000E-01
        X( 2) =  .50000000E-01
        X( 3) =  .10139413E+01
        X( 4) =  .99376532E+00
        INEQUALITY CONSTRAINTS      OCCURRING AT
        G( 1) =  .57321249E-01      SAMPLE POINT  1
        G( 2) =  .99904561E-02      SAMPLE POINT  2
        G( 3) =  .10014581E-01      SAMPLE POINT  3
        G( 4) =  .41908093E-01      SAMPLE POINT  4
        G( 5) =  .16324419E+00      LOWER BOUND   1
        G( 6) =  .13536168E+00      UPPER BOUND   1
        G( 7) =  .14407706E+00      LOWER BOUND   2
        G( 8) =  .15654641E+00      UPPER BOUND   2
        G( 9) =  .20077066E+01      EXTRA CONST   1
NUMBER OF FUNCTION EVALUATIONS =    620
            FOLLOWING IS THE OPTIMUM SOLUTION
        Z( 1) =  .50000000E-01
        Z( 2) =  .50000000E-01
        Z( 3) =  .10139413E+01
        Z( 4) =  .99376532E+00
```

Fig. 6 [continued]. Results of the discrete optimization.

EFFICIENT CALCULATION OF EXACT GROUP DELAY SENSITIVITIES

J.W. BANDLER and M.R.M. RIZK
Group on Simulation, Optimization and Control
McMaster University, Hamilton, Canada

H. TROMP
Laboratory of Electromagnetism and Acoustics
University of Ghent, Ghent, Belgium

ABSTRACT

This paper presents, explicitly, an approach to the exact calculation of group delay and its sensitivities with respect to component parameters based on the adjoint network concept and applicable to linear, time-invariant circuits. In general, no more than four analyses are required and the computational effort is only moderately more than is necessary for a single analysis. The results presented are in a form particularly suited to the computer-aided design of microwave circuits and include useful tables of sensitivity expressions.

INTRODUCTION

The applicability of the adjoint network concept [1,2] to the evaluation of second-order network sensitivities [3,4] has been known for several years. The computation of group delay [5,6] and its sensitivities with respect to component parameters [5] using these ideas has also been suggested. The observation by Temes [5] about the use of perturbation in evaluating group delay sensitivities and the recent implementation by Bandler et al. [7] of perturbation techniques might suggest that exact computation is impractical.

It is the purpose of this paper, therefore, to present explicitly, with the aid of a microwave filter example, a suitable exact approach. In general, no more than four analyses are required and the computational effort is only moderately more than is necessary for a single analysis. The authors are not aware of any similar presentation in the literature.

THEORY

The exact group delay of a linear, time-invariant network N can be evaluated using the adjoint network concept, and requires two network analyses, one of N and one of its adjoint network $\hat{N}$.

The group delay can be defined as [5,6]

$$T_G = -\mathrm{Im} \left\{ \frac{1}{V_0} \frac{\partial V_0}{\partial \omega} \right\} , \tag{1}$$

where V_0 is the output voltage. The sensitivity of T_G w.r.t. the variable parameter ϕ_i will be given by

$$\frac{\partial T_G}{\partial \phi_i} = -\mathrm{Im} \left\{ -\frac{1}{V_0^2} \frac{\partial V_0}{\partial \phi_i} \frac{\partial V_0}{\partial \omega} + \frac{1}{V_0} \frac{\partial^2 V_0}{\partial \phi_i \partial \omega} \right\} . \tag{2}$$

In (2), $\partial V_0/\partial \phi_i$ and $\partial V_0/\partial \omega$ are known from the analyses of N and $\hat{N}$, where $\hat{N}$ is excited only by a current source $\hat{I}_0$ at the output port. Only

---

This work was supported by the National Research Council of Canada under Grant A7239 and by a Graduate Fellowship of the Rotary Foundation to H. Tromp.

$$\frac{\partial^2 V_0}{\partial\phi_i \partial\omega}$$ has to be evaluated.

Second Order Sensitivities

Assume that the elements of N are described by a hybrid matrix, namely,

$$\begin{bmatrix} \underset{\sim}{I}_{aj} \\ \underset{\sim}{V}_{bj} \end{bmatrix} = \begin{bmatrix} \underset{\sim}{Y}_j & \underset{\sim}{A}_j \\ \underset{\sim}{M}_j & \underset{\sim}{Z}_j \end{bmatrix} \begin{bmatrix} \underset{\sim}{V}_{aj} \\ \underset{\sim}{I}_{bj} \end{bmatrix}, \quad j=1,2,\ldots n. \tag{3}$$

The elements of $\hat{N}$ will then be described by [2]

$$\begin{bmatrix} \hat{\underset{\sim}{I}}_{aj} \\ \hat{\underset{\sim}{V}}_{bj} \end{bmatrix} = \begin{bmatrix} \underset{\sim}{Y}_j^T & -\underset{\sim}{M}_j^T \\ -\underset{\sim}{A}_j^T & \underset{\sim}{Z}_j^T \end{bmatrix} \begin{bmatrix} \hat{\underset{\sim}{V}}_{aj} \\ \hat{\underset{\sim}{I}}_{bj} \end{bmatrix}, \quad j=1,2,\ldots,n. \tag{4}$$

Using Tellegen's theorem we may write [2,8]

$$[\underset{\sim}{V}_a^T \; \underset{\sim}{V}_b^T] \begin{bmatrix} \hat{\underset{\sim}{I}}_a \\ \hat{\underset{\sim}{I}}_b \end{bmatrix} - [\underset{\sim}{I}_a^T \; \underset{\sim}{I}_b^T] \begin{bmatrix} \hat{\underset{\sim}{V}}_a \\ \hat{\underset{\sim}{V}}_b \end{bmatrix} = \sum_{j=1}^{n} [\underset{\sim}{V}_{aj}^T \; \underset{\sim}{V}_{bj}^T] \begin{bmatrix} \hat{\underset{\sim}{I}}_{aj} \\ \hat{\underset{\sim}{I}}_{bj} \end{bmatrix} - \sum_{j=1}^{n} [\underset{\sim}{I}_{aj}^T \underset{\sim}{I}_{bj}^T] \begin{bmatrix} \hat{\underset{\sim}{V}}_{aj} \\ \hat{\underset{\sim}{V}}_{bj} \end{bmatrix}, \tag{5}$$

where we assume an unindexed equation of the form of (3) describes the complete network with subscript a denoting voltage excited ports and b current excited ports. Applying the linear operator $\partial^2/\partial\phi\partial\psi$ to (5), where ϕ and ψ are variable parameters, and taking $\underset{\sim}{V}_a$ and $\underset{\sim}{I}_b$ as discrete and independent, we have [3,4,8]

$$\frac{\partial^2 \underset{\sim}{V}_b^T}{\partial\phi\partial\psi} \hat{\underset{\sim}{I}}_b - \frac{\partial^2 \underset{\sim}{I}_a^T}{\partial\phi\partial\psi} \hat{\underset{\sim}{V}}_a = \sum_{j=1}^{n} \begin{bmatrix} \dfrac{\partial^2 \underset{\sim}{V}_{aj}^T}{\partial\phi\partial\psi} & \dfrac{\partial^2 \underset{\sim}{V}_{bj}^T}{\partial\phi\partial\psi} \end{bmatrix} \cdot \begin{bmatrix} \hat{\underset{\sim}{I}}_{aj} \\ \hat{\underset{\sim}{I}}_{bj} \end{bmatrix} - \sum_{j=1}^{n} \begin{bmatrix} \dfrac{\partial^2 \underset{\sim}{I}_{aj}^T}{\partial\phi\partial\psi} & \dfrac{\partial^2 \underset{\sim}{I}_{bj}^T}{\partial\phi\partial\psi} \end{bmatrix} \begin{bmatrix} \hat{\underset{\sim}{V}}_{aj} \\ \hat{\underset{\sim}{V}}_{bj} \end{bmatrix}. \tag{6}$$

Differentiating (3) w.r.t. ψ and ϕ, and substituting the result appropriately into (6) we obtain

$$\frac{\partial^2 \underset{\sim}{V}_b^T}{\partial\phi\partial\psi} \hat{\underset{\sim}{I}}_b - \frac{\partial^2 \underset{\sim}{I}_a^T}{\partial\phi\partial\psi} \hat{\underset{\sim}{V}}_a = \sum_{j=1}^{n} \begin{bmatrix} \dfrac{\partial^2 \underset{\sim}{V}_{aj}^T}{\partial\phi\partial\psi} & \dfrac{\partial^2 \underset{\sim}{I}_{bj}^T}{\partial\phi\partial\psi} \end{bmatrix} \begin{bmatrix} \hat{\underset{\sim}{I}}_{aj} \\ -\hat{\underset{\sim}{V}}_{bj} \end{bmatrix}$$

$$+ \sum_{j=1}^{n} \Bigg\{ [\underset{\sim}{V}_{aj}^T \; \underset{\sim}{I}_{bj}^T] \begin{bmatrix} \dfrac{\partial^2 \underset{\sim}{Y}_j^T}{\partial\phi\partial\psi} & \dfrac{\partial^2 \underset{\sim}{M}_j^T}{\partial\phi\partial\psi} \\ \dfrac{\partial^2 \underset{\sim}{A}_j^T}{\partial\phi\partial\psi} & \dfrac{\partial^2 \underset{\sim}{Z}_j^T}{\partial\phi\partial\psi} \end{bmatrix} + \begin{bmatrix} \dfrac{\partial \underset{\sim}{V}_{aj}^T}{\partial\phi} & \dfrac{\partial \underset{\sim}{I}_{bj}^T}{\partial\phi} \end{bmatrix} \begin{bmatrix} \dfrac{\partial \underset{\sim}{Y}_j^T}{\partial\psi} & \dfrac{\partial \underset{\sim}{M}_j^T}{\partial\psi} \\ \dfrac{\partial \underset{\sim}{A}_j^T}{\partial\psi} & \dfrac{\partial \underset{\sim}{Z}_j^T}{\partial\psi} \end{bmatrix}$$

$$+\begin{bmatrix}\frac{\partial \underset{\sim}{V}_{aj}^T}{\partial\psi} & \frac{\partial \underset{\sim}{I}_{bj}^T}{\partial\psi}\end{bmatrix}\begin{bmatrix}\frac{\partial \underset{\sim}{Y}_{j}^T}{\partial\phi} & \frac{\partial \underset{\sim}{M}_{j}^T}{\partial\phi}\\ \frac{\partial \underset{\sim}{A}_{j}^T}{\partial\phi} & \frac{\partial \underset{\sim}{Z}_{j}^T}{\partial\phi}\end{bmatrix} + \begin{bmatrix}\frac{\partial^2 \underset{\sim}{V}_{aj}^T}{\partial\phi\partial\psi} & \frac{\partial^2 \underset{\sim}{I}_{bj}^T}{\partial\phi\partial\psi}\end{bmatrix}\begin{bmatrix}\underset{\sim}{Y}_{j}^T & \underset{\sim}{M}_{j}^T\\ \underset{\sim}{A}_{j}^T & \underset{\sim}{Z}_{j}^T\end{bmatrix}\Bigg\}\begin{bmatrix}-\hat{\underset{\sim}{V}}_{aj}\\ \hat{\underset{\sim}{I}}_{bj}\end{bmatrix}. \qquad (7)$$

Using (4) which defines the adjoint network $\hat{N}$, (7) will be

$$\frac{\partial^2 \underset{\sim}{V}_{b}^T}{\partial\phi\partial\psi}\hat{\underset{\sim}{I}}_b - \frac{\partial^2 \underset{\sim}{I}_{a}^T}{\partial\phi\partial\psi}\hat{\underset{\sim}{V}}_a = \sum_{j=1}^{n}\Bigg\{[\underset{\sim}{V}_{aj}^T\ \ \underset{\sim}{I}_{bj}^T]\begin{bmatrix}\frac{\partial^2 \underset{\sim}{Y}_{j}^T}{\partial\phi\partial\psi} & \frac{\partial^2 \underset{\sim}{M}_{j}^T}{\partial\phi\partial\psi}\\ \frac{\partial^2 \underset{\sim}{A}_{j}^T}{\partial\phi\partial\psi} & \frac{\partial^2 \underset{\sim}{Z}_{j}^T}{\partial\phi\partial\psi}\end{bmatrix} + \begin{bmatrix}\frac{\partial \underset{\sim}{V}_{aj}^T}{\partial\phi} & \frac{\partial \underset{\sim}{I}_{bj}^T}{\partial\phi}\end{bmatrix}\begin{bmatrix}\frac{\partial \underset{\sim}{Y}_{j}^T}{\partial\psi} & \frac{\partial \underset{\sim}{M}_{j}^T}{\partial\psi}\\ \frac{\partial \underset{\sim}{A}_{j}^T}{\partial\psi} & \frac{\partial \underset{\sim}{Z}_{j}^T}{\partial\psi}\end{bmatrix}$$

$$+\begin{bmatrix}\frac{\partial \underset{\sim}{V}_{aj}^T}{\partial\psi} & \frac{\partial \underset{\sim}{I}_{bj}^T}{\partial\psi}\end{bmatrix}\begin{bmatrix}\frac{\partial \underset{\sim}{Y}_{j}^T}{\partial\phi} & \frac{\partial \underset{\sim}{M}_{j}^T}{\partial\phi}\\ \frac{\partial \underset{\sim}{A}_{j}^T}{\partial\phi} & \frac{\partial \underset{\sim}{Z}_{j}^T}{\partial\phi}\end{bmatrix}\Bigg\}\begin{bmatrix}-\hat{\underset{\sim}{V}}_{aj}\\ \hat{\underset{\sim}{I}}_{bj}\end{bmatrix}. \qquad (8)$$

To evaluate (8) first-order sensitivities of voltages and currents of element j with respect to ϕ and ψ have to be found. Considering the ports of the jth element as the ports of interest we can find the first-order sensitivities needed. Examining (8) we find that the hybrid matrix of the element j is differentiated with respect to ϕ and with respect to ψ. If the parameter ϕ does not belong to the jth element the derivative of the hybrid matrix with respect to ϕ will be zero. The same condition occurs with the parameter ψ. Assuming that each element has one parameter, k additional analyses of $\hat{N}$ will be needed to find (8), where k is the number of variable parameters.

Group Delay Sensitivities

Considering a single output voltage V_0 and setting the adjoint voltage excitation vector $\underset{\sim}{V}_a$ to zero, and replacing the parameter ϕ by ϕ_i and ψ by ω, (8) will be

$$\frac{\partial^2 V_0}{\partial\phi_i\partial\omega}\hat{I}_0 = \sum_{j=1}^{n}[\underset{\sim}{V}_{aj}^T\ \ \underset{\sim}{I}_{bj}^T]\begin{bmatrix}\frac{\partial^2 \underset{\sim}{Y}_{j}^T}{\partial\phi_i\partial\omega} & \frac{\partial^2 \underset{\sim}{M}_{j}^T}{\partial\phi_i\partial\omega}\\ \frac{\partial^2 \underset{\sim}{A}_{j}^T}{\partial\phi_i\partial\omega} & \frac{\partial^2 \underset{\sim}{Z}_{j}^T}{\partial\phi_i\partial\omega}\end{bmatrix}\begin{bmatrix}-\hat{\underset{\sim}{V}}_{aj}\\ \hat{\underset{\sim}{I}}_{bj}\end{bmatrix}$$

$$+\sum_{j=1}^{n}\begin{bmatrix}\frac{\partial \underset{\sim}{V}_{aj}^{T}}{\partial\phi_i} & \frac{\partial \underset{\sim}{I}_{bj}^{T}}{\partial\phi_i}\end{bmatrix}\begin{bmatrix}\frac{\partial \underset{\sim}{Y}_j^{T}}{\partial\omega} & \frac{\partial \underset{\sim}{M}_j^{T}}{\partial\omega}\\ \frac{\partial \underset{\sim}{A}_j^{T}}{\partial\omega} & \frac{\partial \underset{\sim}{Z}_j^{T}}{\partial\omega}\end{bmatrix}\begin{bmatrix}-\hat{\underset{\sim}{V}}_{aj}\\ \hat{\underset{\sim}{I}}_{bj}\end{bmatrix} \tag{9}$$

$$+\sum_{j=1}^{n}\begin{bmatrix}\frac{\partial \underset{\sim}{V}_{aj}^{T}}{\partial\omega} & \frac{\partial \underset{\sim}{I}_{bj}^{T}}{\partial\omega}\end{bmatrix}\begin{bmatrix}\frac{\partial \underset{\sim}{Y}_j^{T}}{\partial\phi_i} & \frac{\partial M_j^{T}}{\partial\phi_i}\\ \frac{\partial \underset{\sim}{A}_j^{T}}{\partial\phi_i} & \frac{\partial \underset{\sim}{Z}_j^{T}}{\partial\phi_i}\end{bmatrix}\begin{bmatrix}-\hat{\underset{\sim}{V}}_{aj}\\ \hat{\underset{\sim}{I}}_{bj}\end{bmatrix} .$$

The first term in (9) can be found from the currents and voltages in N and $\hat{N}$. Note that ω is common, in general, to all the elements.

Let us define $\underset{\sim}{E}_j$ to be

$$\underset{\sim}{E}_j \triangleq \begin{bmatrix}\frac{\partial \underset{\sim}{Y}_j}{\partial\omega} & \frac{\partial \underset{\sim}{A}_j}{\partial\omega}\\ \frac{\partial \underset{\sim}{M}_j}{\partial\omega} & \frac{\partial \underset{\sim}{Z}_j}{\partial\omega}\end{bmatrix} . \tag{10}$$

Introduce now a new network $\hat{N}'$, which is the same as $\hat{N}$, but excited at the ports of each element by current and voltage sources $\hat{\underset{\sim}{I}}'^{s}_{aj}$ and $\hat{\underset{\sim}{V}}'^{s}_{bj}$, where

$$\begin{bmatrix}-\hat{\underset{\sim}{I}}'^{s}_{aj}\\ \hat{\underset{\sim}{V}}'^{s}_{bj}\end{bmatrix} \triangleq \underset{\sim}{E}_j^{T}\begin{bmatrix}-\hat{\underset{\sim}{V}}_{aj}\\ \hat{\underset{\sim}{I}}_{bj}\end{bmatrix} . \tag{11}$$

Consequently, by conventional adjoint network theory

$$\sum_{j=1}^{n}\begin{bmatrix}\frac{\partial \underset{\sim}{V}_{aj}^{T}}{\partial\phi_i} & \frac{\partial \underset{\sim}{I}_{bj}^{T}}{\partial\phi_i}\end{bmatrix}\begin{bmatrix}-\hat{\underset{\sim}{V}}'^{s}_{aj}\\ \hat{\underset{\sim}{I}}'^{s}_{bj}\end{bmatrix} = G_i' , \tag{12}$$

where G_i' is the sensitivity component of N w.r.t. ϕ_i, known in terms of the voltages and currents in N and $\hat{N}'$. The second term in (9) is now G_i'.

Next, consider a network N', which is the same as N, but excited at the ports of each element by current and voltage sources $\underset{\sim}{I}'^{s}_{aj}$ and $\underset{\sim}{V}'^{s}_{bj}$, given by

$$\begin{bmatrix}\underset{\sim}{I}'^{s}_{aj}\\ \underset{\sim}{V}'^{s}_{bj}\end{bmatrix} = \underset{\sim}{E}_j\begin{bmatrix}\underset{\sim}{V}_{aj}\\ \underset{\sim}{I}_{bj}\end{bmatrix} , \tag{13}$$

so that in N'

$$\begin{bmatrix}\underset{\sim}{I}'_{aj}\\ \underset{\sim}{V}'_{bj}\end{bmatrix} = \begin{bmatrix}\underset{\sim}{I}'^{s}_{aj}\\ \underset{\sim}{V}'^{s}_{bj}\end{bmatrix} + \begin{bmatrix}\underset{\sim}{Y}_j & \underset{\sim}{A}_j\\ \underset{\sim}{M}_j & \underset{\sim}{Z}_j\end{bmatrix}\begin{bmatrix}\underset{\sim}{V}'_{aj}\\ \underset{\sim}{I}'_{bj}\end{bmatrix} . \tag{14}$$

Differentiating (3) w.r.t. ω gives

$$\begin{bmatrix} \dfrac{\partial \underset{\sim}{I}_{aj}}{\partial \omega} \\ \\ \dfrac{\partial \underset{\sim}{V}_{bj}}{\partial \omega} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \underset{\sim}{Y}_{j}}{\partial \omega} & \dfrac{\partial \underset{\sim}{A}_{j}}{\partial \omega} \\ \\ \dfrac{\partial \underset{\sim}{M}_{j}}{\partial \omega} & \dfrac{\partial \underset{\sim}{Z}_{j}}{\partial \omega} \end{bmatrix} \begin{bmatrix} \underset{\sim}{V}_{aj} \\ \\ \underset{\sim}{I}_{bj} \end{bmatrix} + \begin{bmatrix} \underset{\sim}{Y}_{j} & \underset{\sim}{A}_{j} \\ \\ \underset{\sim}{M}_{j} & \underset{\sim}{Z}_{j} \end{bmatrix} \begin{bmatrix} \dfrac{\partial \underset{\sim}{V}_{aj}}{\partial \omega} \\ \\ \dfrac{\partial \underset{\sim}{I}_{bj}}{\partial \omega} \end{bmatrix}. \qquad (15)$$

Comparing (14) and (15), we see that

$$\begin{bmatrix} \dfrac{\partial \underset{\sim}{V}_{aj}}{\partial \omega} \\ \\ \dfrac{\partial \underset{\sim}{I}_{bj}}{\partial \omega} \end{bmatrix} = \begin{bmatrix} \underset{\sim}{V}'_{aj} \\ \\ \underset{\sim}{I}'_{bj} \end{bmatrix} \qquad (16)$$

so that the third term in (9) can be calculated from the currents and voltages in N' and N̂.

This means that the network response, its sensitivities, the group delay and its sensitivities can be evaluated by at most four network analyses, independently of the number of variable parameters. This approach is useful for networks with a large number of parameters.

Table I shows $\underset{\sim}{E}_j$ for some elements and for two possible hybrid-matrix formulations. Note that the first-order sensitivity components w.r.t. ω used in the calculation of the group delay, can be found by multiplying $\underset{\sim}{E}_j$ by appropriate vectors of the voltages and currents in N and N̂. Table II shows second-order sensitivity expressions needed in evaluating the first term of (9).

EXAMPLE

Consider the filter shown in Fig. 1 with parameter values as shown [9]. Table III gives a comparison of the group delay sensitivities with respect to characteristic impedances and delay times obtained by this method and perturbation. The parameters were perturbed by an absolute value of 0.5×10^{-7} both ways and using quadratic interpolation (central differences). The appropriate excitations of N, N̂, N' and N̂' are indicated in Fig. 1.

CONCLUSIONS

An efficient approach to the exact calculation of group delay sensitivities is presented, based on the adjoint network concept. A microwave filter example was used to demonstrate explicitly the analyses required. In nodal analysis, we can take advantage of the fact that $\partial \underset{\sim}{Y}/\partial \omega$ has the same structure as the nodal admittance matrix $\underset{\sim}{Y}$ along with LU factorization [10], which needs to be performed only once. In cascaded network analysis we use the adjoint network method to find Thevenin and Norton equivalents [11] prior to carrying out the second analysis of the original and adjoint networks. Further details of the theory, its interpretation and implementation are in [12], as well as tables of special cases of the entries to Tables I and II when $\omega\tau = \pi/2$.

REFERENCES

[1] S.W. Director and R.A. Rohrer, "Automated network design -- the frequency domain case", IEEE Trans. Circuit Theory, vol. CT-16, Aug. 1969, pp. 330-337.

[2] J.W. Bandler and R.E. Seviora, "Current trends in network optimization", IEEE Trans. Microwave Theory Tech., vol. MTT-18, Dec. 1970, pp. 1159-1170.

[3] G.A. Richards, "Second-derivative sensitivity using the concept of the adjoint network", Electronics Letters, vol. 5, Aug. 1969, pp. 398-399.

[4] J.W. Bandler and R.E. Seviora, "Wave sensitivities of networks", IEEE Trans. Microwave Theory Tech., vol. MTT-20, Feb. 1972, pp. 138-147.

[5] G.C. Temes, "Exact computation of group delay and its sensitivities using adjoint-network concept, "Electronics Letters, vol. 6, July 1970, pp. 483-485.

[6] J.W. Bandler, "Computer-aided circuit optimization", in Modern Filter Theory and Design, G.C. Temes and S.K. Mitra, Eds. New York: Wiley-Interscience, 1973.

[7] J.W. Bandler, J.R. Popović and V.K. Jha, "Cascaded network optimization program", IEEE Trans. Microwave Theory Tech., vol. MTT-22, Mar. 1974, pp. 300-308.

[8] P. Penfield, Jr., R. Spence and S. Duinker, "A generalized form of Tellegen's theorem", IEEE Trans. Circuit Theory, vol. CT-17, Aug. 1970, pp. 302-305.

[9] J.W. Bandler, C. Charalambous, J.H.K. Chen and W.Y. Chu, "New results in the least pth approach to minimax design", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-80, Mar. 1975.

[10] S.W. Director, "LU factorization in network sensitivity calculations", IEEE Trans. Circuit Theory, vol. CT-18, Jan. 1971, pp. 184-185.

[11] S.W. Director and D.A. Wayne, "Computational efficiency in the determination of Thevenin and Norton equivalents", IEEE Trans. Circuit Theory, vol. CT-19, Jan. 1972, pp. 96-98.

[12] M.R.M. Rizk, "Efficient simulation of circuits for amplitude and delay optimization in the frequency domain, "Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-102, July 1975.

TABLE I

ELEMENT DERIVATIVES WITH RESPECT TO FREQUENCY

Element	$\frac{\partial \underset{\sim}{Y}}{\partial \omega}$	$\frac{\partial \underset{\sim}{Z}}{\partial \omega}$
inductance	$-\frac{1}{j\omega^2 L}$	jL
capacitance	jC	$-\frac{1}{j\omega^2 C}$
short-circuited lossless transmission line[a]	$jY\tau \csc^2 \omega\tau$	$jZ\tau \sec^2 \omega\tau$
open-circuited lossless transmission line[a]	$jY\tau \sec^2 \omega\tau$	$jZ\tau \csc^2 \omega\tau$
lossless transmission line[a]	$-jY\tau \csc \omega\tau \begin{bmatrix} -\csc \omega\tau & \cot \omega\tau \\ \cot \omega\tau & -\csc \omega\tau \end{bmatrix}$	$jZ\tau \csc \omega\tau \begin{bmatrix} \csc \omega\tau & \cot \omega\tau \\ \cot \omega\tau & \csc \omega\tau \end{bmatrix}$

[a] For transmission lines, Z is the characteristic impedance, Y the characteristic admittance and τ the delay time.

TABLE II
SECOND-DERIVATIVE SENSITIVITY EXPRESSIONS

Element	$\frac{\partial^2 \underset{\sim}{Y}}{\partial\phi_i \partial\omega}$	$\frac{\partial^2 \underset{\sim}{Z}}{\partial\phi_i \partial\omega}$	ϕ_i
inductance	$\frac{1}{j\omega^2 L^2}$	j	L
capacitance	j	$\frac{1}{j\omega^2 C^2}$	C
short-circuited lossless transmission line[a]	$-jY^2\,\tau\,\csc^2\omega\tau$	$j\tau\sec^2\omega\tau$	Z
	$j\tau\csc^2\omega\tau$	$-jZ^2\tau\sec^2\omega\tau$	Y
	$jY\csc^2\omega\tau\,(1-2\,\omega\tau\cot\omega\tau)$	$jZ\sec^2\omega\tau\,(1+2\,\omega\tau\tan\omega\tau)$	τ
open-circuited lossless transmission lines[a]	$-jY^2\,\tau\,\sec^2\omega\tau$	$j\tau\csc^2\omega\tau$	Z
	$j\tau\sec^2\omega\tau$	$-jZ^2\tau\csc^2\omega\tau$	Y
	$jY\sec^2\omega\tau\,(1+2\,\omega\tau\tan\omega\tau)$	$jZ\csc^2\omega\tau\,(1-2\,\omega\tau\cot\omega\tau)$	τ
lossless transmission line[a]	$jY^2\,\tau\csc\omega\tau\begin{bmatrix}-\csc\omega\tau & \cot\omega\tau\\ \cot\omega\tau & -\csc\omega\tau\end{bmatrix}$	$j\,\tau\csc\omega\tau\begin{bmatrix}\csc\omega\tau & \cot\omega\tau\\ \cot\omega\tau & \csc\omega\tau\end{bmatrix}$	Z
	$-j\tau\csc\omega\tau\begin{bmatrix}-\csc\omega\tau & \cot\omega\tau\\ \cot\omega\tau & -\csc\omega\tau\end{bmatrix}$	$-jZ^2\tau\csc\omega\tau\begin{bmatrix}\csc\omega\tau & \cot\omega\tau\\ \cot\omega\tau & \csc\omega\tau\end{bmatrix}$	Y
	$-jY\csc\omega\tau\left\{\begin{bmatrix}-\csc\omega\tau & \cot\omega\tau\\ \cot\omega\tau & -\csc\omega\tau\end{bmatrix} -\omega\tau\begin{bmatrix}-2\csc\omega\tau\cot\omega\tau & \cot^2\omega\tau+\csc^2\omega\tau\\ \cot^2\omega\tau+\csc^2\omega\tau & -2\csc\omega\tau\cot\omega\tau\end{bmatrix}\right\}$	$jZ\csc\omega\tau\left\{\begin{bmatrix}\csc\omega\tau & \cot\omega\tau\\ \cot\omega\tau & \csc\omega\tau\end{bmatrix} -\omega\tau\begin{bmatrix}2\csc\omega\tau\cot\omega\tau & \csc^2\omega\tau+\cot^2\omega\tau\\ \csc^2\omega\tau+\cot^2\omega\tau & 2\csc\omega\tau\cot\omega\tau\end{bmatrix}\right\}$	τ

TABLE III
COMPARISON OF EXACT GROUP DELAY SENSITIVITIES WITH THOSE OBTAINED BY QUADRATIC INTERPOLATION

Normalized frequency		0.5	0.6	0.7	0.8	0.9	1.0
Group delay n sec		2.4864451	0.1803433	0.89523951	0.78200823	0.71710741	0.71408924
$\frac{\partial T_G}{\partial Z_1} = \frac{\partial T_G}{\partial Z_7}$	exact	1.4736349	0.29951487	-0.11744568	0.14143858	0.13616490	-0.09274814
	perturbation	1.4736349	0.29951486	-0.11744568	0.14143858	0.13616490	-0.09274814
$\frac{\partial T_G}{\partial Z_2} = \frac{\partial T_G}{\partial Z_6}$	exact	-5.9194980	-1.0742097	-0.38574124	-0.33653188	-0.29702668	-0.22950730
	perturbation	-5.9194980	-1.0742096	-0.38574124	-0.33653189	-0.29702668	-0.22950731
$\frac{\partial T_G}{\partial Z_3} = \frac{\partial T_G}{\partial Z_5}$	exact	3.9441187	0.45417754	0.28130585	0.14502516	0.12701400	0.15618995
	perturbation	3.9441187	0.45417754	0.28130585	0.14502516	0.12701401	0.15618237
$\frac{\partial T_G}{\partial Z_4}$	exact	-12.313434	-1.2170421	-0.52517215	-0.72479982	-0.49985720	-0.38232839
	perturbation	-12.313434	-1.2170421	-0.52517214	-0.72479983	-0.49985720	-0.38232840
$\frac{\partial T_G}{\partial \tau_1} = \frac{\partial T_G}{\partial \tau_7}$	exact	-1.4354808	0.22931632	1.5821739	0.40019946	1.5881769	1.1275706
	perturbation	-1.4354806	0.22931645	1.5821737	0.40019959	1.5881770	1.1275697
$\frac{\partial T_G}{\partial \tau_2} = \frac{\partial T_G}{\partial \tau_6}$	exact	-22.015614	-3.2718017	-0.35544272	-0.11376393	-0.55456595	0.60609634
	perturbation	-22.015616	-3.2718019	-0.35544258	-0.11376406	-0.55456595	0.60609648
$\frac{\partial T_G}{\partial \tau_3} = \frac{\partial T_G}{\partial \tau_5}$	exact	-34.385645	-2.8021411	-1.7929873	0.16147974	1.6499029	0.98148156
	perturbation	-34.385645	-2.8021414	-1.7929873	0.16147974	1.6499030	0.95919596
$\frac{\partial T_G}{\partial \tau_4}$	exact	-34.417731	-2.3702256	0.60008825	-1.2475054	-0.84413713	0.78227899
	perturbation	-34.417732	-2.3702259	0.60008825	-1.2475054	-0.84413713	0.78227885

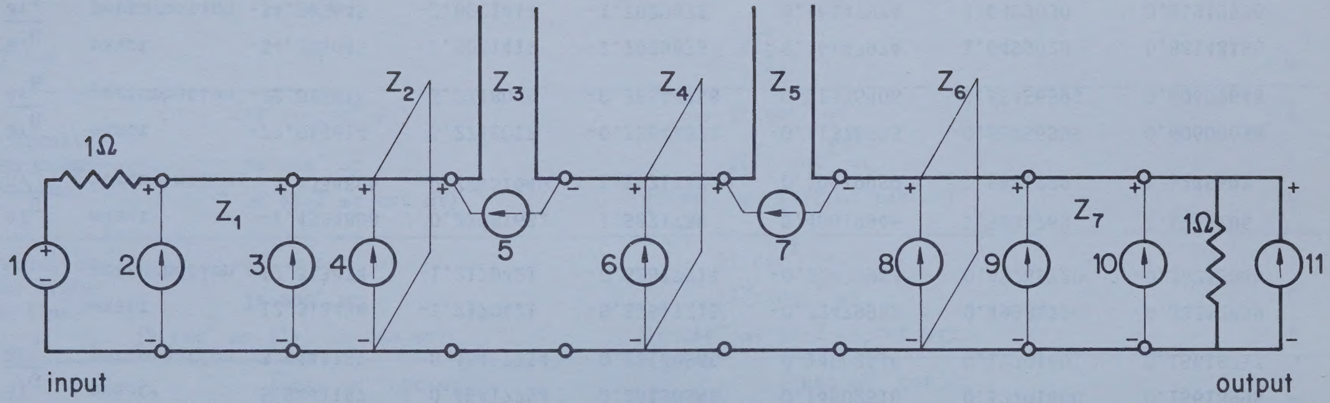

1Ω
Z2
Z3
Z4
Z5
Z6
Z1
Z7
1
2
3
4
5
6
7
8
9
10
11
1Ω
input
output

Fig. 1. Seven-section bandpass filter with sources required for the analyses. For analysis of N source 1 is V_g; for analysis of $\hat{N}$ source 11 is $\hat{I}_0$; remaining sources are 0. For analysis of N' and $\hat{N}'$ sources 1 and 11 are zero; the remaining sources are, respectively, obtained from (13) and (11). All sections are quarterwave at 2.175 GHz. $Z_1 = Z_7 = 0.606595$, $Z_2 = Z_6 = 0.303547$, $Z_3 = Z_5 = 0.722287$, $Z_4 = 0.235183$.

ELEMENT COMBINATIONS FOR SYSTEMATIC MACROMODELING OF INTEGRATED CIRCUITS*

L. C. McAFEE, JR.
Electron Physics Laboratory
Department of Electrical & Computer Engineering
The University of Michigan
Ann Arbor, Michigan 48104

ABSTRACT

A systematic macromodeling procedure is presented in this paper. The approach illustrates the computational efficiency gained by using pre-solved element combinations whose external connections have been specified. The elementary combinations have resulted in improved computational efficiencies of 1.6 to 2.4. The results for a TTL NAND gate structure will highlight the application of these systematic macromodeling techniques.

I. INTRODUCTION

Electronic technology has developed to the state where the design of circuits and systems involves a large and growing use of integrated circuit (IC) components. The capability of IC technology to fabricate and use IC components in design has surpassed the capability for efficient modeling and computer-aided analysis of systems containing these components. This divergence in capabilities has provided the stimulus for new research efforts to update IC modeling technology via macromodeling [1-5]. Two goals of macromodeling are (1) to represent IC components as discrete elements and (2) to maintain reasonable accuracy of model and computational efficiency of computer-aided-analysis algorithms. Existing macromodeling efforts have relied strongly on the knowledge and modeling abilities of the engineer, and not on systematic algorithmic automated approaches. This paper describes some preliminary research efforts that looked at an approach to provide systematic methods for arriving at macromodels. This study was a feasibility study to see if efficiencies could be gained by looking at element combinations.

The approach to be described takes advantage of the topology of combinations of elements that are constructed from normal discrete elements (resistors, diodes, transistors, etc.). Only the characteristics of the external terminals of the element combinations are seen by a network analysis algorithm. The advantage to be gained is that the internal solution of these element combinations has been pre-solved for maximum computational efficiency.

Element combinations that have been studied are: series and parallel diodes, resistor-diode combinations, resistor-transistor combinations, etc. These combinations while retaining similar matrix set-up times have led to significantly improved matrix solution times; the pre-solved internal solution achieves a reduced node count which leads to the reduced matrix solution times. This paper will present the formulation of some element combinations and show how and why the improved efficiencies are realized.

---

* This work was supported by a research grant from the Division of Research Development and Administration, The University of Michigan.

II. EXPONENTIAL FUNCTION APPROXIMATION

Another factor that significantly improved efficiency was the use of an algorithm that approximated the exponential function, rather than using the standard Fortran supplied exponential library function. This approximation turned out to be 2.2 times more efficient that the Fortran supplied function, and it had less than one percent error. The motivation for pursuing this effort is that an IC has more junctions than it has any other element. Thus time saved in computing the exponential function will be repaid many times.

The approach was to pre-solve and store values of the exponential function for integer argument values. The exponential values at other arguments were obtained by using a polynomial approximation between the nearest two pre-solved pre-saved values. The polynomial approximation was a cubic equation between the two nearest function values. The exponential approximation used is

$$\mathrm{EXPAPR} = (1.0 + 1.35914{*}U - 0.5{*}U^2 + 0.85914{*}U^3)\ \mathrm{EXPL} \qquad (1)$$

where EXPL is the pre-solved pre-saved exponential value evaluated at the nearest integer argument xL that is less than the desired argument x, and U = x - xL. This function assumes that the argument is positive; negative valued arguments are computed by taking the reciprocal of the EXPAPR function value above.

III. ELEMENT COMBINATIONS

Some element combinations that have been studied to provide computational analysis efficiencies will now be discussed. The details of developing the general algorithm will be shown for the resistor-diode combination first; the development of the other combinations will follow a similar formulation with most details omitted.

For the development of the element combinations, it is assumed that node analysis is the method used to set up the mathematical equations describing the network. The formulations will be such that the element combination of concern is the only element in the network other than current sources. Thus any interaction of the element combination with other elements is avoided. Newton's method is assumed to be the numerical method for nonlinear iteration. Table I lists the symbols used throughout this paper.

III.A. RESISTOR-DIODE COMBINATION

MATRIX ANALYSIS:

A series resistor-diode combination is shown in Fig. 1. The diode is modeled by

$$\mathrm{ID} = \mathrm{IS}[\exp(\lambda{*}\mathrm{VD}) - 1] \qquad (2)$$

where VD = V2 - V3. The incremental conductance GD of the diode at the iterated diode current is given by

$$\mathrm{GD} = \lambda(\mathrm{ID} + \mathrm{IS}) \quad . \qquad (3)$$

The node equations are

$$f1 = -I + G(V1 - V2) \tag{4a}$$

$$f2 = -G(V1 - V2) + IS[\exp(\lambda*(V2 - V3)) - 1] \quad . \tag{4b}$$

The set-up of the matrix equation for Newton's method is

$$\begin{bmatrix} G & -G \\ -G & G+GD \end{bmatrix} \begin{bmatrix} DV1 \\ DV2 \end{bmatrix} = \begin{bmatrix} -f1 \\ -f2 \end{bmatrix} \quad . \qquad \begin{matrix}(5a)\\(5b)\end{matrix}$$

The computational effort to set up this matrix equation is the following. The equation f1 uses 1 M, equation f2 uses the resistive current from f1 plus 1 exp + 2 Ms, GD uses 1 M; thus the total set-up effort is 1 exp + 4 Ms. To solve the matrix equation of 2x2 size requires 3 Ms + 3 Ds. Thus the total computational effort to analyze the resistor and diode network is ops = 1 exp + 7 Ms + 3 Ds = 27.6 Ms. The approximate number of multiplications is based on the comparative execution times for the IBM 370/168 CPU as listed in Table II.

ELEMENT COMBINATION ANALYSIS:

As an element combination, this resistor-diode element must be modeled using only two terminals and having one mathematical model component relation. The approach reported here is to represent the combination as having nodes 1 and 3 as the external terminals with node 2 internal to the element. The values of the voltage at the external terminals are considered as unknowns, while the value of the voltage at the internal node is symbolically pre-solved and computed as part of the component model evaluation. Thus the two considerations of the component model are the terminal I-V characteristic and the evaluation of the internal node voltage value. Another consideration (at this point) is that the Newton iteration voltage changes based on the element combination model should be the same as those based on matrix analysis [Eq. (5)].

There are two equivalent viewpoints for developing an algorithm for evaluating the value of the internal node. First, the mathematical model can be used which retains the computational methods flavor. Second, the companion model can be used which retains the network flavor [6]. The equivalence of these two approaches is readily obvious since the companion model is a linearized network developed from the mathematical equation model. Only the mathematical model approach will be developed here due to space limitations.

MATHEMATICAL MODEL: Since it is desired that these techniques be applicable to large IC networks, the computation of the internal node voltage value(s) should be based on automated algorithms. The structuring of the internal node evaluation around a mathematical model is in search of that ultimate end. The following development shows a systematic approach to arrive at both the internal node evaluation and the element combination terminal V-I mathematical model.

Going back to the original matrix equation for Newton's iteration, Eq. (5), the second equation is a linearization of the equation for node 2, whereas the first equation is the equation for node 1. The equation for node 2 is used to compute the internal node 2 value, and the equation for node 1 is used to represent the element combination in conjunction with the solution from the second equation.

The second equation of the matrix is

$$-G\ DV1 + (G + GD)DV2 = -\ G(V2 - V1) - ID \quad . \qquad (6)$$

Using that

$$DVm = V,m{+}1 - Vm \qquad (7)$$

the second equation leads to

$$\begin{aligned} V2 = VD,m{+}1 &= [G\ V1,m{+}1 - (IDm - GDm\ VDm)]/G + GD) \\ &= (G\ V1,m{+}1 - CURm)/(G + GD) \quad . \qquad (8) \end{aligned}$$

But this equation is the same as that obtained by solving the matrix equation, Eq. (5). Thus equivalence is achieved.

The component model for the element combination is then the first node equation, Eq. (4a), which with Eq. (8) gives

$$f1 = -I + G[V1\,GD + CURm]/(G + GD) \quad . \qquad (9)$$

The equation to solve for the element combination solution is (using Newton's method)

$$[G\ GD/(G + GD)]DV1 = -f1 \quad . \qquad (10)$$

The solution of this equation for DV1 agrees with the solution of the matrix equation for DV1m; thus the element combination model is equivalent to the matrix equation analysis when based on the iterated changes in the node voltages.

From this systematic mathematical approach, both the internal node evaluation, Eq. (8), and the component model, Eq. (9), have been obtained. A systematic approach is particularly needed when larger systems are being analyzed and/or when more than one internal node is to be used and evaluated.

OPERATIONS COUNT: The solution for VD,m+1 requires 1 M + 1 D, f1 (Eq. (4a)) uses 1 M, the Jacobian uses 1 M + 1 D, ID uses 1 exp + 2 Ms, GD uses 1 M, and the solution for DV1 uses 1 D. Thus the total effort is ops = 1 exp + 6 Ms + 3 Ds = 26.6 Ms for a ratio compared with the Newton's matrix formulation of 27.6/26.6 = 1.04. Using EXPAPR the element combination count is reduced by approximately nine Ms (See Table II), thus the overall ratio is 27.6/17.6 = 1.57. A minimal effect in improving the computational efficiency is due to this combination; EXPAPR has the significant effect.

III.B. SERIES DIODES

Two not identical diodes in series are shown in Fig. 2. The diodes are described by

$$ID1 = IS1\,[\exp(\lambda 1 * VD1) - 1] \qquad (11a)$$

$$ID2 = IS2[\exp(\lambda 2*VD2) - 1] \quad . \qquad (11b)$$

For the series diodes in Fig. 2, the diode voltages are given by

$$VD1 = V1 - V2, \; VD2 = V2 - V3 \quad . \qquad (12)$$

For the remainder of this paper, diodes will be labeled as D1 and D2, etc., will have characteristics as in Eq. (11), and the diode conductances will be calculated using Eq. (3).

For the series diodes element combination, the internal node voltage is computed as (knowing the external node voltages V1 and V3)

$$V2 = [GD1*(V1-VD1) + GD2*(V3+VD2)]/(GD1 + GD2) \quad . \qquad (13)$$

Only one of the two diode currents need be calculated since ID1=ID2=ID. The component model chosen in this case was

$$ID = IS1[\exp(\lambda 1*(V1 - V2)) - 1] \quad . \qquad (14)$$

The network node equation and Jacobian are

$$f1 = -I + ID \quad , \qquad (15a)$$

$$J = GD1\ GD2/(GD1 + GD2) \quad . \qquad (15b)$$

The element combination is used to take advantage of the fact that KCL must be satisfied at the internal node, thus saving one diode evaluation. This is a deviation from the restriction that the iterated voltage changes must be the same for element combination analysis as for matrix analysis, as was invoked for the resistor-diode combination.

A comparison of the computational effort to analyze series diodes is the following. Using standard matrix node analysis techniques, the set up and solution effort is equivalent to approximately 46.3 Ms. Using the element combination formulation, the calculation for V2 requires 2 Ms plus 1 D, the diode current evaluation requires 2 Ms + 1 exp, GD1 and GD2 again require 2 Ms, and the Jacobian requires 2 Ms + 1 D. The solution for V1 requires 1 D. Therefore, the total operations count is ops = 1 exp + 8 Ms + 3 Ds = 28.6 Ms. The ratio of comparison is approximately 46.3/28.6 = 1.6. Using EXPAPR the ratio increases to 46.3/19.6 = 2.4.

IDENTICAL SERIES DIODES: If the diodes are identical, then even greater efficiencies can be realized. The constraints on identical diodes are that

$$IS = IS1 = IS2, \; \lambda = \lambda 1 = \lambda 2 \quad . \qquad (16)$$

For these identical diodes in series, further constraints are

$$ID = ID1 = ID2, \; VD1 = VD2 = V/2 = V1/2 \quad . \qquad (17)$$

Thus the current through the two diodes is

$$ID = IS(\exp(\lambda*V/2) - 1.0) \quad . \qquad (18)$$

The computational savings is that only one exponential evaluation need be done. The expected savings is approximately a factor of 2 normally, and approximately a factor of 4 using EXPAPR. This form will give the same iterated voltage changes as standard matrix node analysis.

III.C. PARALLEL DIODES

Parallel diodes have several possible applications. A typical application is Schottky TTL where a Schottky diode, in parallel with the base-collector junction, is used to prevent saturation of the BJT's and thus improve the propagation delay of the gate. Another application is the input of a TTL gate where the input is modeled as multiple BJT's rather than as a multi-emitter transistor [7]. A third application is the I^2L configuration where the PNP injector/load and the NPN driver are modeled as two BJT's rather than as one I^2L structure [8].

With parallel diodes both diodes D1 and D2 have the same junction voltage. The characteristics of D1 and D2 are given in Eq. (11). Parallel diodes provide no inherent efficiencies as an element combination because there is no internal node. Thus no reduction in unknown count is achieved when considering parallel diodes. The only computational efficiency possible is that provided by EXPAPR.

Using normal node analysis, the matrix to be solved is

$$(GD1 + GD2)\ DV1 = -f1 = -\ (ID1 + ID2) \quad . \tag{19}$$

The total computational effort count is 4 Ms + 2 exp for the diode currents plus 2 Ms for the diode conductances, and 1 D for DV1. Thus the total computational effort count is ops = 2 exp + 6 Ms + 1 D = 40.7 Ms. With the EXPAPR approximation, the operation count reduces to = 22.7 Ms for a ratio of improvement of 40.7/22.7 = 1.8.

IDENTICAL PARALLEL DIODES: Additional efficiency can be achieved when the parallel diodes are identical, Eq. (16). The common circuit variable is

$$V = VD1 = VD2 \quad . \tag{20a}$$

The sum of the currents into the diodes is

$$ID = IS1(\exp(\lambda * V) - 1) + IS2(\exp(\lambda * V) - 1)$$

$$= 2\ IS(\exp(\lambda * V) - 1) \quad . \tag{20b}$$

Just as with series identical diodes, the diodes can be treated as one diode with overall computational savings factors of approximately 2 to 4.

III.D. RESISTOR-BJT COMBINATION

The resistor-transistor combination studied is one having a base lead-in resistor as shown in Fig. 3. This combination occurs for numerous circuits. The three external terminals are nodes 1, 2, and 3 with the internal node 4. The BJT model is the Ebers-Moll static resistive model. Using matrix node analysis, the operation count for the three-equation system for nodes 1, 3, and 4 is ops = 2 exp + 22 Ms + 6 Ds = 63.2 Ms.

For the element combination, the currents are computed as

$$IB = G(V1 - V4) \tag{21a}$$

$$ID1 = IS1[\exp(\lambda 1*VD1) - 1] \tag{21b}$$

$$ID2 = [(IB - ID1(1 - AF)]/(1 - AR) \quad . \tag{21c}$$

The current ID2 is calculated by invoking KCL at node 4. Knowing the values of the external node voltages, then node 4 is given by

$$DV4 = [G\ DV1+GD11\ DV2+GD21\ DV3]/(G+GD11+GD21) \tag{22a}$$

$$V4 = V4m + DV4 \ , \tag{22b}$$

where GD11 = (1-AF)*GD1, and GD21 = (1-AR)*GD2. The set up requires 1 M each for GD1, GD11, GD2, and GD21, 4 Ms + 1 D for V4, 1 exp + 2 Ms for ID1, 1 M + 1 D for ID2, 1 M for f3, and 6 M + 2 D for the remaining matrix set-up for a total of 1 exp + 19 Ms + 4 Ds. The solution of a 2x2 matrix equation requires 3 Ms + 3 Ds. Thus the total effort for the element combination formulation is ops = 1 exp + 22 Ms + 7 Ds = 47.8 Ms, for a ratio of 63.2/47.8 = 1.3. If EXPAPR is used, the equivalent effort is ops = 38.8 Ms for a ratio of 63.2/38.8 = 1.6.

This combination could possibly lead to faster convergence since the internal node is not being iterated in the sense that KCL is always satisfied at that node. It is entirely possible that this element may replace the BJT element due to its enhanced efficiency. Thus whenever a BJT model is needed, this combination would be used instead.

III.E. TTL NAND GATE: AN ELEMENT COMBINATION EXAMPLE

A TTL NAND gate has been macromodeled using these methods. It was determined that a factor of 1.1 was realized just by representing the IC gate as an entity rather than a group of separate elements. This is because the elements are in a prespecified and anticipated order, rather than a random order of elements as in an arbitrary network which must be decoded.

The input transistor for the TTL gate is a resistor-BJT combination. The upper transistor and the diode in the totem-pole output stage are effectively a series diode combination, since the upper BJT never saturates for small betas (<13) and can be modeled by an active region model.

When the element combinations were constructed and EXPAPR used, an additional factor of 1.8 was realized. Thus the overall improvement in speed due to using the element combinations and EXPAPR to model the TTL NAND gate entity was approximately 1.98. EXPAPR had a significant effect on this result since there are 10 junctions to be evaluated in the TTL gate.

IV. SUMMARY AND CONCLUSIONS

The motivation for this study was to find an approach suitable for systematic macromodeling of ICs. This study was a feasibility study to see if computational efficiencies could be gained by looking at element combinations (lower level macromodels). Computational advantages of 1.6 to 2.4 have been achieved when using these element combinations and EXPAPR. Table III lists the results in summary form. The resistor-resistor combination is for two series resistances. With element combinations, the matrix solution time is significantly improved due to a reduced equation/unknown count (the result of a reduced node count). These improvements have been obtained with simple element combinations. It is anticipated that even greater gains can be realized when larger and higher level element combinations are constructed. Also, a greater proportion of junctions will likely be on larger ICs, making the effect of EXPAPR more noticeable.

Future efforts will be devoted to obtaining element combinations when capacitive elements are included. Also, studies will concentrate on developing approaches suitable for element combinations having more than one internal node.

REFERENCES

1. N. B. Rabbat, W. D. Ryan, and S. Q. A. M. A. Hossain, "A Computer Modeling Approach for LSI Digital Structures," IEEE Trans. On Electron Devices, vol. ED-22, pp. 523-531, August 1975.
2. N. B. Rabbat, W. D. Ryan, and S. Q. A. M. A. Hossain, "Transient Analysis of Multiple-Input Integrated Digital Structures," IEEE Trans. On Electron Devices, vol. ED-20, pp. 496-508, May 1973.
3. G. R. Boyle, B. M. Cohn, D. O. Pederson, and J. E. Solomon, "Macromodeling of Integrated Circuit Operational Amplifiers," IEEE J. Solid-State Circuits, vol. SC-9, No. 6, pp. 353-364, December 1974.
4. J. R. Greenbaum, "Modeling of Operational Amplifiers in Computer-Aided Design," Gen. Elec. Tech. Inf. Series, No. R69EMH2, February 1969.
5. J. R. Greenbaum and W. A. Miller, "Digital-IC Models for Computer-Aided Design, Parts 1-4:...," Electronics, December 6, 1973 to February 7, 1974.
6. D. A. Calahan, Computer Aided Network Design, Rev. Ed., McGraw-Hill: New York, 1972.
7. H. C. Lin, "DC Analysis of Multiple Collector and Multiple Emitter Transistors in Integrated Circuits," IEEE J. Solid-State Circuits, vol. SC-4, pp. 20-24, February 1969.
8. H. H. Berger and S. K. Wiedmann, "Terminal-Oriented Model for Merged Transistor Logic," IEEE J. Solid-State Circuits, vol. SC-9 No. 5, pp. 211-217, October 1974.

TABLE I. LIST OF SYMBOLS

SYMBOL	MEANING
BJT	bipolar junction transistor
CURm	IDm - GDm VDm
D,Ds	division(s)
DV	iterated change in voltage
EC	element combination
exp	exponent function
EXPAPR	approximation to the exponential function
f1,f2,...,	node equations describing network
GD	junction diode incremental conductance
IC	integrated circuit
ID	junction diode current
IS	reverse saturation current for junction diode
λ	voltage coefficient in diode function
m	nonlinear iteration counter
M,Ms	multiplication(s)
ops	operations count
V	node voltage
VD	junction diode voltage

TABLE II. IBM 370/168 CPU TIMES

FUNCTION	TIME (μSEC)	EQUIVALENT MULTIPLICATIONS
mult	1.975	1.0
div	2.558	1.3
exp	33.	16.7
EXPAPR	15.3	7.7

TABLE III. SUMMARY OF SAVINGS

(ops in terms of equivalent multiplications)

		DUE TO ELEMENT COMBINATIONS ONLY		DUE TO BOTH ELEMENT COMBS. AND EXPAPR	
ELEMENT COMB.	MATRIX OPS	EC OPS	RATIO OF IMPROVEMENT	EC(2) OPS	OVERALL RATIO
RES-RES	6.9	3.3	2.1	3.3	2.1
RES-DIO	27.6	26.6	1.04	17.6	1.6
SER-DIO	46.3	28.6	1.6	19.6	2.4
PAR-DIO	40.7	40.7	1.0	22.7	1.8
RES-BJT	63.2	47.8	1.3	38.8	1.6

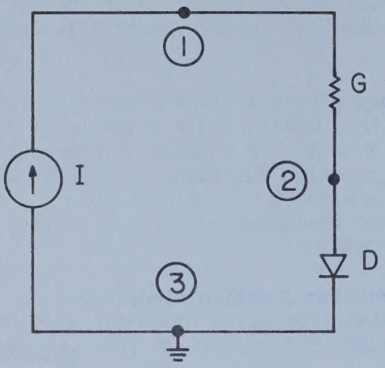


FIG. 1 RESISTOR-DIODE COMBINATION

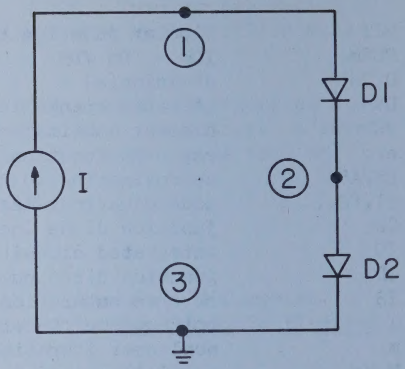


FIG. 2 SERIES-DIODE COMBINATION

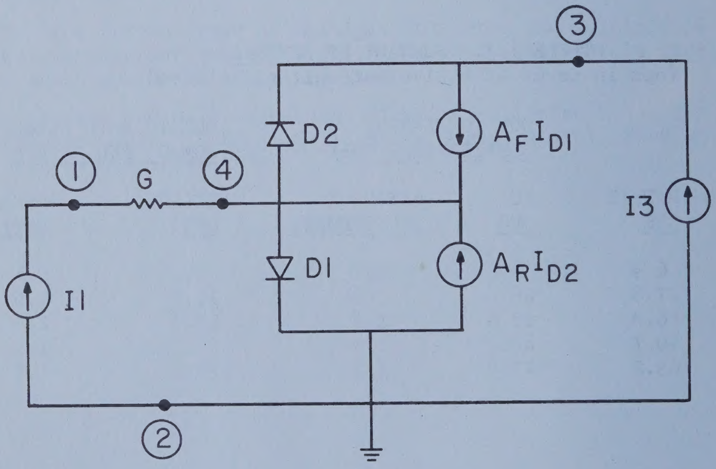

③
D2
$A_F I_{D1}$
①
G
④
I3
D1
$A_R I_{D2}$
I1
②

FIG. 3 RESISTOR-TRANSISTOR COMBINATION

COMPUTER-AIDED DESIGN OF BROAD-BAND MICROWAVE CLASS-C TRANSISTOR AMPLIFIERS

F. FILICORI, V.A. MONACO
Istituto di Elettronica, Università di Bologna, Italy

ABSTRACT

A computer-oriented technique for designing microwave class-C transistor amplifiers is described. This is accomplished by computing amplifier performance in terms of transistor large-signal S-parameters, measured for different input powers and minimizing a specification-related objective function by means of Fletcher-Powell's algorithm. Possibilities and limitations are discussed comparing computed with measured performance.

INTRODUCTION

This paper describes a computer technique for designing input- and output-matching networks of a single-stage microwave class-C transistor amplifier. It operates by means of an objective function related to the required amplifier performance. The specifications may be flatness of insertion gain vs frequency at a given available generator power, or a given variation of the same insertion gain vs the available generator power at a given frequency, or both.

The optimization program follows Fletcher-Powell's minimization technique (1). Constraints on the component parameters, which restrict the range of the admitted nominal values, are introduced by the addition of a penalty function to the objective function, as in the Fiacco-McCormick approach (2).

The circuit analysis is performed by the SCAMAT program which determines the scattering matrix of a given n-port in terms of its component parameters (3,4). Large-signal S-parameters (5,6) are used to characterize the power transistor and are considered as functions of the transistor input power instead of the average dc collector current, in order to account for transistor non-linearity automatically. This allows the analysis program to choose the appropriate set of transistor parameters for every operating condition, and to determine the amplifier performance.

Moreover, a description is given of the technique followed for measuring transistor large-signal S-parameters and the problems involved are discussed. The performances of a microwave class-C transistor amplifier implemented in accordance with the results of the optimization are shown and compared with the given specifications.

THE OPTIMIZATION PROGRAM

The optimization computer program written for designing the input- and output-matching networks of a microwave class-C transistor amplifier operates by finding the minimum of an objective function whose most general form is:

$$F = \sum_{i,j} W_{ij} \left[G(f_i,P_j) - \overline{G}(f_i,P_j) \right]^2 + H \sum_i \left[V_P(f_i,P_M) - \overline{V} \right]^2 + \sum_k g_k$$

The first summation, extended to all the frequencies f_i and all the available generator powers P_j, takes account of the differences between the actual insertion gain $G(f_i,P_j)$ and the required $\overline{G}(f_i,P_j)$. Making an opportune choice of the weights W_{ij} the right compromise between the given specifications of the insertion gain vs frequency and the insertion gain vs available generator power can be reached.

The second summation is introduced to ensure that, at the maximum available generator power P_M, the collector voltage V_P is near the value $\overline{V}$ which prevents transistor saturation with maximum collector efficiency (5). The weight H properly accounts for the transistor saturated output power specification.

The third summation represents a penalty function which limits the component values within an admitted range. The term relative to the k^{th} component takes the form:

$$g_k = C_k \left[L_s \left(\frac{X_k - X_k^{max}}{\Delta X_k} \right)^4 + L_i \left(\frac{X_k - X_k^{min}}{\Delta X_k} \right)^4 \right]$$

The coefficients L_s and L_i are zero except in the regions:

$X_k > X_k^{max}$ where $L_s = 1$,

$$X_k < X_k^{min} \qquad \text{where } L_i = 1,$$

so that g_k is zero in the admitted range $X_k^{min} \leq X_k \leq X_k^{max}$ and increases as shown in Fig. 1, reaching the value C_k when $X_k = X_k^{max} + \triangle X_k$ or $X_k = X_k^{min} - \triangle X_k$.

The optimization program follows the Fletcher-Powell minimization technique (1) and does not present particular convergency problems when the starting point is suitably chosen.

The insertion gain G and the collector voltage amplitude V_P are computed, in terms of the network component parameters X_k, by means of a specially written analysis routine described in the following section.

ANALYSIS OF THE CLASS-C AMPLIFIER

The class-C amplifier analysis is made considering the transistor S-parameters to be dependent on input power so that, for each value of the available generator power P_j, the appropriate set of transistor parameters is chosen. This is carried out by means of a routine which operates step by step according to the algorithm shown in Fig. 2.

At a given step N a value $P = P^{(N-1)}$ is assumed for the transistor input power and the corresponding set of transistor parameters $S_T(P)$ is chosen. Using the SCAMAT program the scattering matrix of the circuit is computed and the transistor input power $P^{(N)}$ is determined in terms of the available generator power P_j and the circuit S-parameters. A comparison between the computed input power $P^{(N)}$ and the previously assumed value $P^{(N-1)}$ allows for verification of the assumption's validity. If the difference between the computed and the assumed values is not less than the admitted error, then the set of S-parameters corresponding to $P^{(N)}$ is assumed to be the new value and the procedure repeated until the test is satisfied. The chosen transistor S-parameters are thus appropriate for the given available generator power P_j, and the circuit S-matrix computed by SCAMAT describes the amplifier completely, allowing for the evaluation of the network functions for the given frequency and available generator power. The insertion gain coincides, as is known, with the transmission parameter between the input and output ports. Transistor input power $P(f_i, P_j)$ is computable in terms of the available generator power P_j and of the incident and reflected waves at the transistor input port. To this end a dummy directional coupler DC_1, which does not modify incident and reflected waves at the transistor input port, while making them available at ports 3 and 4 of the network respectively, has been inserted, as shown in Fig. 3. In such conditions the transistor input power for a given available generator power P_j is:

$$P = (S_{31}^2 - S_{41}^2)P_j$$

A second dummy directional coupler DC_2 makes available at ports 6 and 5 the incident and reflected waves at transistor output; thus the amplitude of the rf transistor collector voltage V_P in the second term of the objective function can be determined by:

$$V_P = \sqrt{2}\,(S_{51} + S_{61})\,\sqrt{R_0 \cdot P_j}$$

S-parameters being normalized with respect to the generator internal resistance R_0.

TRANSISTOR CHARACTERIZATION

The use of large-signal S-parameters for designing microwave class-C transistor amplifiers has been proposed by a number of authors (5,6). Leighton et al. reported that owing to package parasitics, waveforms at a microwave class-C operated transistor are fairly sinusoidal so that linear methods, if applied with care, may be used successfully. In fact, measurements of large-signal S-parameters effected with different input and output circuit impedances showed good agreement if the collector dc current was the same for both cases. In conclusion they suggested considering S-parameters as functions of collector dc current.

Starting from these criteria, in order to make the automatic design of the amplifier possible, we tried to relate S-parameters to a rf variable directly computable using an analysis program. The most significant one appeared to be the transistor input power, on which the collector dc current certainly depends. Following such a hypothesis, measurements have been effected for different values of transistor input power. Moreover, in order to verify that these parameters do not depend on test circuit impedance, measurements have been made in both a 50 ohm and an 8 ohm system for different transistors. This showed that the S-parameters normalized with respect to the same resistance are almost the same at a given transistor input power, whatever the measurement system impedance.

However, it must be pointed out that measurements in a low impedance system, though more troublesome, make it possible to operate with higher input power without collector saturation, as well as requiring less generator available power, owing to better transistor input matching. In Fig. 4 the test set for large-signal S-parameter measurements is shown. It consists of an S-parameter test set, a microwave power generator, a power meter and a vector voltmeter. Two 30 dB attenuators were inserted at the inputs of the vector voltmeter to allow higher power operation. The power meter was connected by means of a power splitter to the reference channel in order to measure the power incident at the transistor. The two impedance transformers TR_1 and TR_2 are inserted when low impedance measurements must be effected. In this case the S-matrix of the two-port containing the transistor and the two transformers is the object of the measurements. The knowledge of the S-parameters of the transformers allows easy computation of the S-matrix of the transistor alone. In Fig. 5 the amplitude and the phase of S_{11} and S_{21} of an HP 35853B transistor at f = 2 GHz vs transistor input power P_{IN} are plotted.

Parameters S_{12} and S_{22} were measured class-A biasing the transistor since, in a class-C operation, removing the generator from the input would cause no dc collector current to flow and so no meaningful data could be obtained. These parameters for the tested transistors were also found to be almost independent of collector dc current. They were therefore considered to be functions of the frequency alone, in agreement with the results of previous authors. The maximum allowed collector voltage V_P was determined by the expression:

$$V_P = V_{cc} - V_{sat}$$

V_{cc} being the collector supply voltage and V_{sat} the rf saturation voltage measured by plotting output power vs collector voltage at a given input power as suggested in paper (5).

EXPERIMENTAL RESULTS

Following the proposed technique an amplifier, whose specifications were P_{out} = 0.5 W and G = 6 dB in the band 1.8 ÷ 2 GHz, was designed. The transistor chosen was an HP 35853B whose large-signal S-parameters were measured in a 50 ohm system by the test set in Fig. 4 for different input powers and frequencies. For input- and output-matching network topology a double stub solution was chosen.

The starting values for the lengths of the lines (all 50 ohm) were chosen by a specially written routine which operates by first computing, according to a narrow-band technique, the source and load impedances of the transistor for obtaining the specified output power and insertion gain at the middle band frequency.

The subsequent optimization led to a satisfying broad band performance by changing the line lengths and characteristic impedances, as shown in Table 1. The designed amplifier (see Fig. 6) was implemented on a teflon substrate. The physical dimensions of the strip-lines corresponding to the electrical lengths and characteristic impedances obtained by optimization were computed by the ASSAM program (7). The computed and measured power gains of the amplifier in Fig. 6 are plotted vs frequency and vs available generator power in Fig. 7. Dot-dash lines and solid lines refer to computed values before and after optimization respectively; dashed lines refer to measured values.

CONCLUSIONS

A procedure for designing broad-band class-C transistor amplifiers was proposed. An example of application involving the implementation of a micro-strip prototype was then given. A comparison between measured and computed values showed the validity of the technique described, at least for the implemented amplifier whose transistor had a transition frequency not much greater than the maximum operating frequency.

REFERENCES

(1) R. Fletcher, M.J.D. Powell: "A rapidly convergent descent method for minimization" - Comput. J., Vol. 6, pp. 163-168, July 1963

(2) A.V. Fiacco, G.P. McCormick: "Nonlinear Programming: Sequential unconstrained minimization techniques" - New York: Wiley, 1968

(3) V.A. Monaco, P. Tiberio: "Automatic scattering matrix computation of microwave circuits" - Alta Frequenza, Vol. 39, pp. 55-64, February 1970

(4) V.A. Monaco, P. Tiberio: "A computer program for circuit analysis from dc to microwave, using scattering parameters" in Dig. Tech. Papers IEEE Symp. Circuit Theory, pp. 119-120, December 1970

(5) W.H. Leighton, R.J. Chaffin, J.G. Webb: "RF amplifier design with large-signal S-parameters" - IEEE Trans. on MTT, Vol. 21, No. 12, pp. 809-814, December 1973

(6) O. Mueller: "Large-signal S-parameter measurement of class-C operated transistors" - Nachrichtentech. Z., p. 64, October 1968

(7) V. Rizzoli: "The calculation of scattering parameters for coupled microstrip arrays of any cross section" - Alta Frequenza, Vol. 42, pp. 191-199, April 1973

Table 1. Electrical lengths and characteristic impedances of lines before and after optimization

	Lengths (el. degrees at 1.9 GHz)						Impedances (ohms)					
Line	1	2	3	4	5	6	1	2	3	4	5	6
Before opt.	150	5	180	90	45	45	50	50	50	50	50	50
After opt.	134	39	71	52	37	32	48	45	48	52	70	53

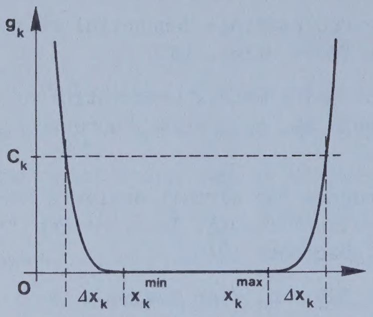


Fig. 1 Contribution g_k to the penalty function due to the parameter X_k.

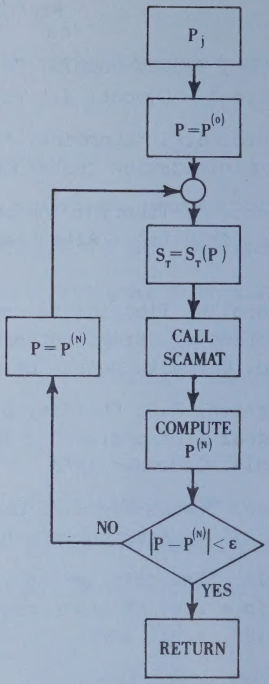

P_j
$P=P^{(o)}$
$S_T=S_T(P)$
CALL SCAMAT
COMPUTE $P^{(N)}$
$|P-P^{(N)}|<\varepsilon$
NO
YES
RETURN
$P=P^{(N)}$

Fig. 2 Flow-chart of the routine which selects the appropriate set of transistor S-parameters for every available generator power.

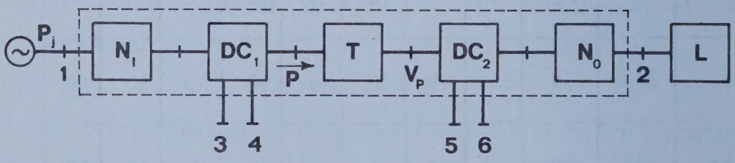

P_j
1
N_I
DC_1
$\vec{P}$
T
V_P
DC_2
N_O
2
L
3
4
5
6

Fig. 3 Block diagram of the analysed amplifier.

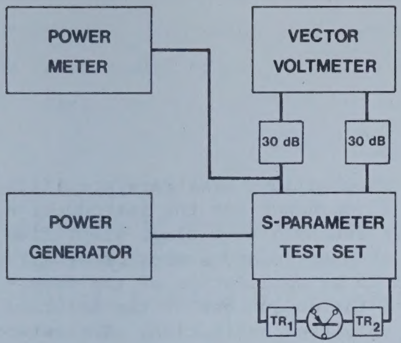

POWER METER
VECTOR VOLTMETER
30 dB
30 dB
POWER GENERATOR
S-PARAMETER TEST SET
TR1
TR2

Fig. 4 Test set for large-signal S-parameter measurements.

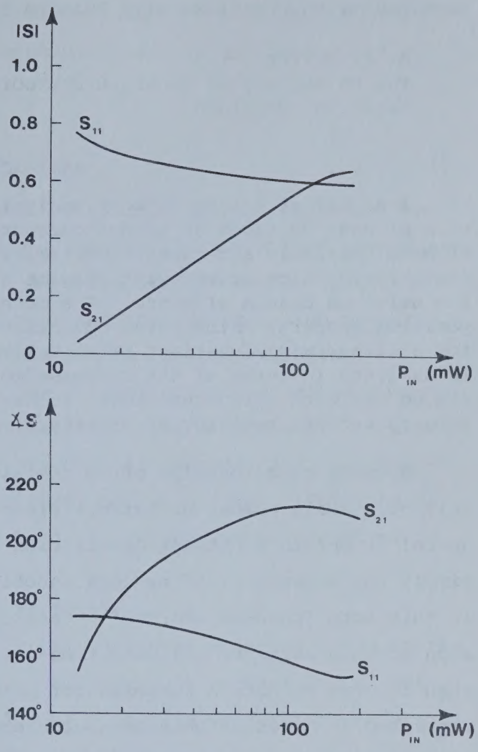


Fig. 5 Amplitude |S| and phase ∡S of parameters S_{11} and S_{21} vs input power P_{IN}, for transistor HP 35853B at f = 2 GHz.

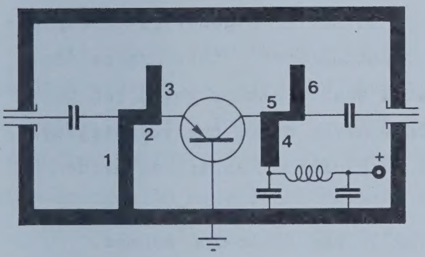


Fig. 6 Strip-line class-C transistor amplifier.

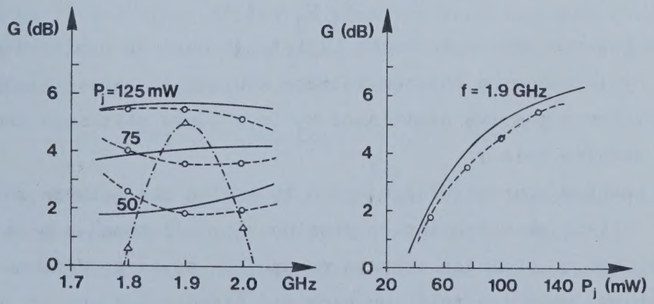


Fig. 7 Computed and measured insertion power gain vs frequency and vs available generator power of the amplifier in Fig. 6. Dashed lines refer to measured values.

SENSITIVITY CALCULATIONS WITH PHANTOM SOURCES

M. H. Miller
The University of Michigan-Dearborn
Dearborn, Michigan

ABSTRACT

A method of linear network analysis which allows arbitrary specification of node voltages or branch currents in exchange for the introduction of more tractable alternative variables is extended to include simplified sensitivity calculations with respect to changes in any number of arbitrarily selected branch elements. The method is an application of the superposition property which takes advantage of the uniqueness of the solution for a linear time-invariant network with specified excitation. The network is analysed in terms of its response to excitation by sources judiciously placed and with strengths chosen to distinguish the individual contribution to network behavior of arbitrarily selected network elements.

Network relationships often are significantly sensitive to variation only of a small number of network elements. In such circumstances it is useful to obtain a network description which displays explicitly and simply the dependence of network functions on selected network components. In this note 'phantom sources'(1) are used to derive a generalized expression of this sort for a linear time-invariant network. This expression then is used to obtain formulas for network sensitivities which can be expressed in terms of phantom source parameters. Hence the versatility of this computational method is extended in a useful manner, as evidenced by an illustrative calculation.

The expression referred to is derived in the following manner. Extract, i.e., select for special emphasis, some (any) number of network branches. Augment each selected branch with a current source so that the combined volt-ampere relation is

$$I_j = Y_j V_j + I_j^* \tag{1}$$

where the phantom source strength is I_j^*. (A dual or hybrid description is obtained by introducing phantom voltage sources in series with branches. For simplicity a passive admittance Y_j is assumed, although the argument does not require this.)

The phantom sources will be used to excite the network so that it displays certain characteristics prominently. Although a particular excitation is used to evaluate certain parameters so as to determine various relationships, once the relationships are established any convenient excitation can be used.

Select the (any) linear function of network variables P that is of interest. Suppose that for some arbitrary (but consistent) choice of phantom source strengths the variable has a value P_x. Distinguish separately as P_{xq} the value of P when the phantom sources are removed, i.e., the 'real' value of P.

Then

$$P_x = P_{xq} + \sum_j T_{xj} I_j^* \qquad (2)$$

where the sum is over however many phantom sources are introduced. The coefficients T_{xj} are network invariants. Equation 2 is a direct consequence of the uniqueness of the network volt-ampere distribution for specified excitation, including initial conditions, and the superposition property.

For the particular purpose intended phantom source strengths are selected to isolate the influence on the network of each selected branch. The prescription for the selection establishes a reference state in which the influence of all of the selected branches is removed, and then measures the influence of each of them in turn, one at a time. (This may be suggestive of the calculation of a hybrid matrix, but there are significant distinctions.)

Select phantom source strengths as follows:

(a) Short circuit, in effect, all the extracted branches by using the phantom sources to carry the necessary short-circuit current, i.e., set $I_j^* = I_{sj}$, $V_j = 0$ for all j. The value of P calculated for this reference condition, which is entirely independent of the extracted branches, is P_{xs}. Substitution in equation 2 obtains the reference equation

$$P_{xs} = P_{xq} + \sum_j I_{sj} T_{xj} \qquad (3)$$

(b) Modify the reference condition by open-circuiting, in effect, a single branch, e.g., $V_j = 0$, $j \neq k$ and $I_k = 0$. To maintain this condition $I_k^* = - Y_k V_{k0}$ where V_{k0} is the open-circuit voltage across Y_k. The short circuit currents in the other extracted branches generally are changed to I_{kj} from the reference values. Designate as P_{xk} the value of P in this condition and substitute in equation 2 to obtain

$$P_{xk} = P_{xq} - Y_k V_{k0} T_{xk} + \sum_{\substack{j \\ j \neq k}} I_{kj} T_{xj} \qquad (4)$$

(c) Repeat the preceding specification for each branch in turn to obtain a set of independent equations of the form of equation 4.

The set of equations (including equation 3) can be summarized as

$$\begin{bmatrix} P_{xs} \\ \hline P^* \end{bmatrix} = \left[\begin{array}{c|c} 1 & U \\ \hline 0 & M \end{array}\right] \begin{bmatrix} P_{xq} \\ \hline IT \end{bmatrix} \qquad (5)$$

where the vector P* has for its $k^{\underline{th}}$ component $P_{xs} - P_{xk}$, IT has for its $k^{\underline{th}}$ component $I_{sk} T_{xk}$, U is an all-unit row vector, and the square matrix M has

has components

$$M_{kj} \overset{\Delta}{=} \begin{cases} 1 + Y_j^*; \ k = j \\ 1 - (I_{jk}/I_{sj}); \ j \neq k \end{cases} \tag{6a}$$

$$Y_j^* \overset{\Delta}{=} Y_j \ (V_{j0}/I_{sj}) \tag{6b}$$

This system of equations is formed by subtracting the reference equation from all the others and factoring the currents.

Note that the admittances of the extracted branches appear explicitly each in its own diagonal component of M, and neither explicitly nor implicitly otherwise.

Inversion of equation 5 obtains

$$P_{xq} = P_{xs} - UM^{-1} \ P^* \tag{7a}$$

$$IT = M^{-1} \ P^* \tag{7b}$$

If a single branch is extracted, equations 7 reduce to an expected bilinear form (2).

$$P_{xq} = \frac{P_{xk} + P_{xs} \ Y_k^*}{1 + Y_k^*} \tag{8a}$$

$$I_{sk} \ T_{xk} = \frac{P_{xs} - P_{xk}}{1 + Y_k^*} \tag{8b}$$

The generalized expression equation 7a has some interest in its own right. Thus once the matrix components have been calculated updating calculations as a result of changes made in the admittance of extracted branches is generally simpler using equation 7 than with a complete reanalysis of the network. The rank of M is likely to be considerably smaller for this calculation than, say, a node admittance matrix.

However, there is another use to which equations 7 can be put. Equation 7a suggests a certain ready availability of the dependence of network variables P_{xq} on the admittance of selected branches, i.e., as it were these admittances stand apart to distinguish their separate contributions. Further, equation 7b suggests that the information available can be expressed in terms of the 'T' coefficients, and these coefficients can be calculated using any set of phantom source strengths which provide a sufficient number of independent equations. As a latter illustration demonstrates this admits some interesting options which simplify calculations.

Assuming that network element variations are confined to the extracted branches differentiation of equation 7a obtains

$$dP_{xq} = -UM^{-1} \ dM \ M^{-1} \ P^* \tag{9}$$

where dM is a diagonal matrix whose j^{th} diagonal component is dY_j^*.

Since P can be any linear variable one choice is the voltage across one of the extracted branches, say V_i. Then for the calculations made in obtaining M there is only a single circumstance for which $V_i \neq 0$, and that is for the open-circuit calculation involving the $k^{\underline{th}}$ branch. Therefore P* had only the single non-zero component $-P_{xk} = -V_{k0}$. Since $P_{xs} = 0$ for this choice of variable it follows that UM^{-1} is a row vector whose $k^{\underline{th}}$ component is V_{kq}/V_{k0}.

With this information, and using equation 7b, equation 9 can be rewritten as

$$dP_{xq} = \sum_k T_{xk} V_{kq} \, dY_k \tag{10}$$

so that

$$\frac{\partial P_{xq}}{\partial Y_k} = T_{xk} V_{kq} \tag{11}$$

Actually equation 11 can be obtained perhaps more easily from equation 8. And from those equations it can be verified that

$$\frac{\partial^2 P_{xq}}{\partial Y_k{}^2} = 2 \frac{\partial P_{xq}}{\partial Y_k} \frac{\partial \ln V_{kq}}{\partial Y_k}$$

$$= 2 T_{xk}{}^{(P)} T_{xk}{}^{(V_k)} V_{kq} \tag{12}$$

where the superscript identifies which variable the coefficient is associated with, e.g., $T_{xk}{}^{(V_k)}$ is the coefficient coupling the variable V_k to $I_k{}^*$ in an equation of the form of equation 2.

The general expression, equation 7a, can be expanded for two extracted branches, from which the interaction sensitivity can be obtained;

$$\frac{\partial^2 P_{xq}}{\partial Y_j \partial Y_k} = \frac{\partial P_{xq}}{\partial Y_j} \frac{\partial \ln V_{jq}}{\partial Y_k} + \frac{\partial P_{xq}}{\partial Y_k} \frac{\partial \ln V_{kq}}{\partial Y_j}$$

$$= T_{xj}{}^{(P)} T_{xk}{}^{(V_{jq})} V_{kq} + T_{xk}{}^{(P)} T_{xj}{}^{(V_{kq})} V_{jq} \tag{13}$$

Figure 1 describes a simplified feedback pair which is used for an illustrative calculation. Although there is not much additional computational effort for each variable considered after the first, calculations are made only for the input conductance (I_0), the voltage gain (V_0), the voltages across the two resistances R_1 and R_2 which are to be 'extracted', and the first and second order sensitivities of I_0 and V_0 to variation of these resistances. The M matrix also is calculated and its convenience discussed.

The two phantom current sources introduced to extract R_1 and R_2 imply three calculations to evaluate P_{xq} and the T coefficients in the equation

$$P_x = P_{xq} + T_{x1} I_1{}^* + T_{x2} I_x{}^* \tag{14}$$

Since any consistent choice for phantom source strengths may be used, a choice offering some computational advantage can be made.

For example choose I_1^* and I_2^* indirectly so that $i = v = 0$, i.e., both stages are deactivated. (Note that nonzero phantom source strengths are a computational fiction not intended for the 'real' network. Hence consistency and not physical reasonableness is the watch-word. This is an important distinction because of disconcerting but quite valid oddities which can occur.) Since $v = 0$, $V_2 = 0$. Note that R_2 is not short-circuited, but rather it carries zero current to conform with Ohm's law. Since $i = 0$ inspection indicates $I_0 = I_2^* = 1/11$. (Units used throughout are to be understood to be amperes, volts, ohms, mhos, as appropriate.) Since there is no current through the base-emitter resistance of the first stage the voltage across the 0.1Ω emitter resistance is $1-I_0$. Therefore there is a a current flow such that $V_0 = -10(1-I_0)$, and to support $V_1 = -11(1-I_0)$ it is necessary to choose $I_1^* = 111(1-I_0)/10$. These values are tabulated (as decimal values) in Table I.

Another convenient choice (indirectly) for phantom source strengths is $v = 0$ but $i \neq 0$; this deactivates (not 'breaks') the feedback loop. For the second of the two 'free' choice choose $I_2^* = 0$ (with a thought ahead toward simplifying the simultaneous equations to be manipulated). Again note that $V_2 = 0$, but R_2 is not short-circuited. In fact the entire current $50i$ flows through the first stage 10Ω shunt feedback resistance. Hence $1 = 51i + 50i\ (10)$ or $i = 1/551$. Other variable values, tabulated in Table I, follow directly.

Finally, for a third excitation, choose (say) $i = 0$, $I_2^* = 0$. Actually almost any choice makes for some simplification, although ingenuity and experience help take advantage of the more favorable opportunities. In this case, since $i = 0$, the voltage across the 0.1Ω emitter resistance of the first stage is $1-I_0$ as before. The feedback loop is deactivated and the analysis is reduced to that of a single stage amplifier with emitter feedback. Variable values calculated are tabulated in Table I.

Table I

Values for selected variables calculated for different phantom source excitations.

	a) $i = v = 0$	b) $v = I_2^* = 0$	c) $i = I_2^* = 0$
I_0	0.0909	0.0926	0.0843
V_0	- 9.0909	- 8.9637	- 9.7506
V_1	-10.0	- 9.8693	-10.6663
V_2	0.	0.	0.0725
I_1^*	10.0909	9.9507	10.2234
I_2^*	0.0909	0.	0.

The three equations obtained on substitution into equation 14 are

$$\begin{bmatrix} P_{xa} \\ P_{xb} \\ P_{xc} \end{bmatrix} = \begin{bmatrix} 1 & 10.0909 & 0.0909 \\ 1 & 9.9507 & 0. \\ 1 & 10.2234 & 0. \end{bmatrix} \begin{bmatrix} P_{xq} \\ T_{x1} \\ T_{x2} \end{bmatrix} \qquad (15)$$

Inversion of the matrix and then substition of P_{xa}, P_{xb}, P_{xc} from Table I for the variables tabulated (others can be added easily) obtains

$$\begin{bmatrix} V_0 \\ I_0 \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} 19.7432 & -2.8849 & 3.0511 \\ 0.3932 & -0.0302 & 0.0285 \\ 19.2042 & -2.9218 & 3.0698 \\ -2.6431 & 0.2656 & -0.4098 \end{bmatrix} \begin{bmatrix} 1 \\ I_1^* \\ I_2^* \end{bmatrix} \qquad (16)$$

Equation 16 presents all the data needed for the calculations now to be made. The first column of the transfer matrix presents the values of the variables for $I_1^* = I_2^* = 0$, i.e., the P_{xq} values.

The second column presents the T coefficient coupling to I_1^*, and the coupling to I_2^* is the third column.

Substitution of the appropriate values from equation 16 into equations 11 through 13 provides the entries for the sensitivities in Table II.

If calculations of this sort are part of an iterative process the use of the M matrix can reduce computational effort. Although the components of the M matrix are derived by choosing particular phantom source excitations that is only an artifice to evaluate the T coefficients in a way which emphasizes certain network properties. The T coefficients are available in equation 16, and that information can be used to evaluate the components of the M matrix.

Table II

Sensitivity calculations for I_0, V_0

(evaluated for $G_1 = 0.1$, $G_2 = 1$ as in Figure 1).

	V_0	I_0
$\frac{\partial}{\partial G_1}$	-55.4025	-0.5802
$\frac{\partial}{\partial G_2}$	- 8.0642	-0.0752
$\frac{\partial^2}{\partial G_1^2}$	323.7461	3.3920
$\frac{\partial^2}{\partial G_2^2}$	6.6046	0.0616
$\frac{\partial^2}{\partial G_1 \partial G_2}$	38.9711	0.3902

For example, to produce the reference condition in which all (two in this case) extracted branches are 'short-circuited' set $V_1 = V_2 = 0$ in equations 16 and calculate the required excitation $I_1^* = -0.6414$ and $I_2^* = -6.8663$; these are the short-circuited currents. Then calculate the various P_{xs}, i.e., $V_0 = 0.6443$ and $I_0 = 0.2172$ in the short-circuit condition.

For the open-circuit condition for R_1 choose $V_1 = 10I_1^*$, i.e., the phantom source and not the network supplies current to R_1. Calculate $I_1^* = 0.0659$, $I_x^* = -6.4077$. The open-circuit voltage across R_1 then is -0.6591 and the short-circuit current through R_2 is I_2^*. Calculate the various P_{x1}, i.e., $V_0 = 0.0027$, $I_0 = 0.2089$.

Finally, for the open circuit condition for R_2 choose $V_2 = -I_2^*$ and calculate $I_1^* = 7.6572$, $I_2^* = 1.0321$, the open-circuit voltage -1.0321 across R_2, $V_0 = 0.8018$, and $I_0 = 0.1912$.

The last three calculations provide all the information needed to calculate the components of M. In fact some of the subsequent manipulation to obtain equations 7 is, in essence, already done. It is only necessary to observe that R_1 and R_2 are involved in these last three calculations only to determine the phantom source current strengths needed to supple the 'open-circuit' current demand of R_1 and R_2 respectively. All the other network values are open-circuit or short-circuit values. Hence, for any R_1 maintain its open-circuit status by choosing $I_1^* = -0.6591/R_F$, where the numerator is the open-circuit voltage. Similarly when R_2 is open-circuited use $I_2^* = -1.0321/R_2$.

With this modification regard the open and short-circuit calculations to correspond to three choices for sets of phantom source strengths involving the extracted resistances explicitly, and these can be used with the general equation 14. Hence, directly,

$$\begin{bmatrix} P_{xs} \\ P_{x1} \\ P_{x2} \end{bmatrix} = \begin{bmatrix} 1 & -0.6414 & -6.8663 \\ 1 & \dfrac{-0.6591}{R_1} & -6.4077 \\ 1 & 7.6572 & \dfrac{-1.0321}{R_2} \end{bmatrix} \begin{bmatrix} P_{xq} \\ T_{x1} \\ T_{x2} \end{bmatrix} \tag{17}$$

This brings back to the status of equation 15 and Table I, except that a different set of excitations is used. The excitations in equation 17 display the total involvement of R_1 and R_2 explicitly. For any choice of R_1 and R_2 network variables and sensitivities can be calculated as described with equation 15. Equation 17 can be useful, for example, in an iterative calculation to adjust R_1 and R_2 to suit some sensitivity criteria.

For a final calculation consider the occasional belated realization that the sensitivity to some network element other than those already extracted is desired. The incremental effort to calculate the additional 'T' coefficient is comparitively small. In some instances what is in effect a source transformation can maneuver an existing phantom source across the new branch to be extracted. If not the independence of the phantom sources (assuming no current cutsets or, if voltage sources are used, voltage loops) simplifies the calculation of the T coefficient for the new source, introduced as needed.

For example, suppose a third phantom current source of strength I_3^* (not shown in Figure 1) shunts the load across which V_0 is measured; the positive sense of the current is taken into ground. Specify i = v = 0 in Figure 1 as a convenient (indirect) excitation. Further, since the third phantom source allows another specification, choose $I_0 = 0$. Then $I_2^* = 0$, and the voltage across the 0.1Ω emitter resistance of the first stage is 1. Therefore $I_1^* = -I_3^* = 10$ for $V_0 = 0$. Substitution in equation 14, extended by a term $T_{x3}I_3^*$, with T_{x3} the only unknown coefficient (for each choice of variable P) evaluates this coefficient. In fact

$$P_x = P_{xq} + 10T_{x1} - 10T_{x3} \qquad (18)$$

with $P_x = 0$ for a choice of I_0, V_0, V_1, and V_2 as the variable. Using matrix entries from equation 16

$$T_{x3}^{(I_0)} = 0.00912$$

$$T_{x3}^{(V_0)} = -0.9106$$

$$T_{x3}^{(V_1)} = -1.0014$$

$$T_{x3}^{(V_2)} = 0.0013$$

Sensitivities with respect to the load resistance can be calculated on substitution into the appropriate formulas.

Actually this belated calculation can be computationally advantageous. After a few phantom sources have been introduced a network generally can be excited so as greatly simplify calculations. Hence, it may be easier overall to obtain the T coefficients in a serial fashion rather than to invert a higher rank matrix.

---

(1) M. H. Miller, "Phantom sources for linear circuit analysis," IEEE Trans. Circuits and Systems, Vol. CAS-22, pp. 698-699, August, 1975.

(2) S. R. Parker, E. Peskin, and P. M. Chirlian, "Application of a bilinear theorum to network sensitivity," IEEE Trans. Circuit Theory (correspondence), Volt. CT-12, pp. 448-450, September, 1965.

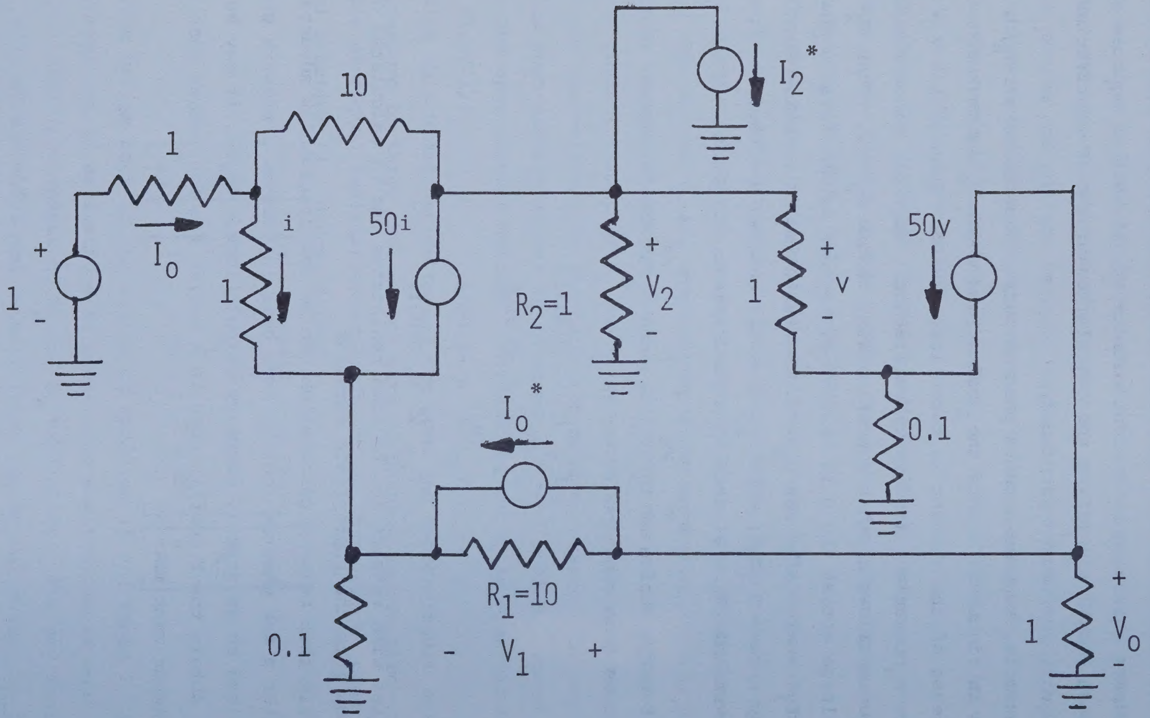

10
1
I0
+
1
-
i
1
50i
I2*
R2=1
+
V2
-
1
+
v
-
50v
0.1
I0*
R1=10
0.1
- V1 +
1
+
V0
-

Figure 1. Feedback pair used for illustrative calculation. Units are ohms, volts, amperes.

ON SOLVABILITY AND SOLUTION OF LINEAR ACTIVE NETWORKS

M. H. MILLER
The University of Michigan-Dearborn
Dearborn, Michigan

Conditions have been derived [1,2,3] which are necessary and sufficient to insure solvability of linear time-invariant networks. In this note similar conditions are derived differently which can be applied in a more versatile manner to determine solvability and obtain the solution in all circumstances in which it exists.

The derivation makes use of the uniqueness of the volt-ampere distribution over a linear time-invariant network with specified excitation (including initial conditions). Each controlled source is replaced by a like independent source. If there is a loop of voltage sources replace one (independent, if any) source by a controlled current source with current control exercized by a sum over branch currents at one of its nodes; this artifice replaces a redundant voltage specification with an accounting for the loop current. Similarly replace one (independent, if any) current source in a current source cutset by a controlled voltage source; control is exercized by a voltage sum over any arc subtended by the source.

A unique linear relationship exists between any network variable y and the strengths of the replacement sources $\underline{u}$;

$$y = y_r + T\underline{u} \tag{1}$$

Apply (1) to the controlling variables $\underline{x}$ of the (replaced) sources

$$\underline{x} = \underline{x}_r + \tilde{T}\underline{u} \tag{2}$$

This equation expresses the network requirements imposed on the relationship between $\underline{x}$ and $\underline{u}$. It is compared for consistency with the requirement

$$\underline{u} = \tilde{G}\underline{x} \tag{3}$$

required if the controlled sources are to be reintroduced; consistency requires

$$(\tilde{U} - \tilde{T}\tilde{G})\underline{x} = \underline{x}_r \tag{4}$$

where $\tilde{U}$ is a unit matrix. A sufficient condition for solvability is nonsingularity of $\tilde{U} - \tilde{T}\tilde{G}$. It is a necessary condition as well, except where $\underline{x}_r = 0$; this condition occurs for a single independent source in a loop or cutset of sources. Interdependence of the controlled sources makes $\tilde{U} - \tilde{T}\tilde{G}$ singular in these circumstances, but a solution may be obtained provided $\tilde{U} - \tilde{T}\tilde{G}$ is not null.

The calculations involved in evaluating $\underline{x}_r$ and $\tilde{T}$ also provide y where a solution exists. If solvability but not a solution is to be determined a simpler procedure is to augment rather than replace controlled sources with independent sources after deactivating specified independent sources. Then y_r and $\underline{x}_r$ are zero in the analysis.

---

[1] W-K Chen and N-T Chan, "On the unique solvability of linear active networks," IEEE Trans. Circuits and Systems, Vol. CAS-21, pp. 26-35, Jan. 1974.

[2] E. J. Purslow, "Solvability and analysis of linear active networks by use of the state equations," IEEE Trans. Circuit Theory, Vol. CT-17, pp. 469-475, Nov. 1970.

[3] N. R. Malik and H. W. Hale, "Equations for active networks; Existence of unique solutions," IEEE Trans. Circuit Theory, Vol. CT-14, pp 37-43, Jan. 1967.

SIMPLIFIED MULTIPLE STUCK-AT-FAULT TEST GENERATION TECHNIQUES

CARL D. LATINO and JON G. BREDESON
Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802

ABSTRACT

A new concept for generating test sets for detecting multiple faults in combinational networks will be introduced in this paper. A procedure which will be called the "Cell Method" allows near optimal test generation with ease of computation for most combinational networks. This procedure involves breaking down complex networks into cells, solving the test generation problem for each cell and then properly integrating these results into the complete test set for the original network.

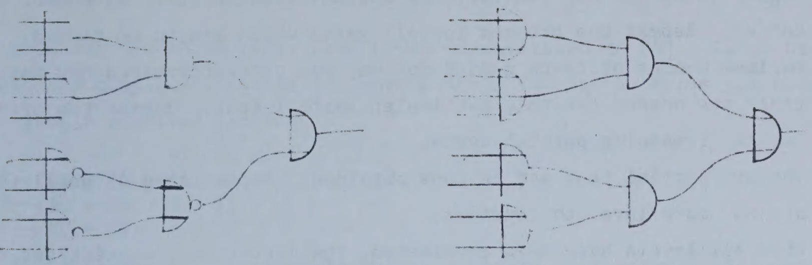
INTRODUCTION

Test generation procedures for multiple fault detection is a problem which has received much attention. The procedures generated to date,[1-7] however, are either tedious or give results which are not minimal. These previous generation procedures considered the network in its original form and attempted to solve the test generation problem on the entire circuit at once.

Since it is easier to generate complete test sets for several small networks separately than for a larger one that has the same total number of gates, a new procedure was attempted. This new procedure allows a calculation which determines what the near minimum number of tests is for a given network, allows the generation of these tests with ease and is applicable to most combinational networks (including networks with reconvergent fanout).

THE CELL METHOD

The Cell Method which will be introduced in this paper is best applied to tree networks free of reconvergent fanout. The best results are obtained by first modifying the network into the equivalent alternating AND/OR form. This will always be possible for tree networks. The two networks in Figure 1 have the same fault properties, however, only network B is in the proper form.



Network A - Tree Network

Network B - Equivalent AND/OR Form

Figure 1

Definition 1: Simultaneously testable - Lines which drive a given gate are simultaneously testable if faults on any of these lines cannot be masked by faults on the other lines. A fault is masked if it can exist undetected.

Definition 2: Individually testable line - When a line feeding a gate is individually testable it implies that a fault existing on this line can be masked by faults on the other lines feeding the same gate.

Definition 3: Desensitizing test - The desensitizing test for a network f is an input pattern D_a $(a = 0,1)$ such that for any fault set α

1. $f^{\alpha}(D_a) = a$, or
2. If $f^{\alpha}(D_a) = \bar{a}$ then $f^{\alpha}(t) = \bar{a}$ for any input pattern t.

   i.e., The desensitizing tests for an AND gate are $D_1 = 111...1$ and $D_o = 000...0$.

The following is the Cell Method.

1) Modify the network if necessary into the alternating AND/OR form
2) Break the network down into cells. Each cell is two level, while cell overlap is limited to one gate.
3) Generate the complete test set for each cell. Include the desensitizing patterns where needed. The complete test set for the output cell is the first partial test set.
4) Expand on the partial test set by examining the cells in the next higher level. If an input gate (of the partial test set) is desensitized, the desensitizing pattern from the overlapping cell must be selected. If a gate (of the partial test set) is being tested for a particular input

pattern, do the following. Determine how many times the pattern for this gate matches the same gate in the appropriate cell of the next higher level (do not consider the desensitizing pattern as one of the tests). Repeat the process for all gates which are to be tested. The maximum number of tests needed for any one gate determines how many tests are needed for this particular partial test. Repeat the process for the remaining partial tests.

5) The new partial test set is thus obtained. Repeat step 4) until there are no more levels to consider.
6) When all levels have been considered, the latest partial test set is the complete test set.

Definition 4: Regular tree network - A tree network is regular if:

a) It is in the alternating AND/OR form
b) Each gate in any given level has the same number of inputs
c) Every input to output path is the same length.

In order to simplify the notation, the binary input pattern to a gate is replaced by its decimal equivalent, i.e., if a gate has inputs 1010, 1100, 0010, etc., they are replaced by 10, 12, 2, respectively.

COMPLETE TEST SET FOR A SINGLE CELL

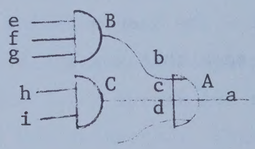


Figure 2

Consider the network in Figure 2. Gate A is completely tested by the input patterns bcd = {000, 001, 010, 100} or using the simplified notation A = {0,1,2,4}. Gate B is completely tested by B = {7,6,5,3} and Gate C is completely tested by C = {3,2,1}. The desensitizing patterns for Gates B and C must be introduced as needed. When Gate A is tested for input pattern "0", lines b, c, and d are simultaneously tested for s-a-1 faults. Since Gate B requires three tests for a zero output, Gate C only two, three tests are needed for testing Gates A, B, and C for s-a-1 faults. These tests may be represented as follows:

		efg	hi	d
A = 0	BC = 62	110	10	0
	BC = 51	101	01	0
	BC = 30	011	00	0

Note that for the third test, Gate C was desensitized to "0". Since Gate C was already completely tested for zero outputs during the first two tests, it no longer required testing.

		efg	hi	d
A = 4	BC = 70	111	00	0
A = 2	BC = 03	000	11	0
A = 1	BC = 00	000	00	1

The six tests detect all possible multiple faults as Theorem 1 will demonstrate.

EXAMPLE

Consider the network in Figure 3. The test set for this network will be generated by the Cell Method.

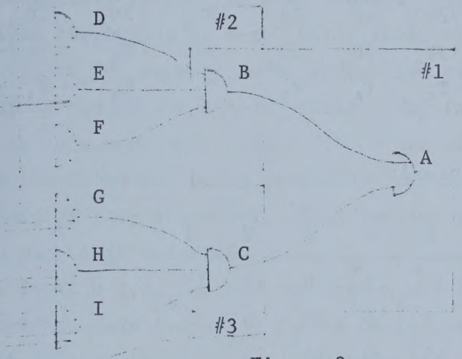


Figure 3

Steps 1) and 2). The cells are labeled in Figure 3.

Step 3) The test sets for each cell are found in a similar manner as the network in Figure 2.

#1	a	b	c	d	e
B	3	5	6	7	0†
C	3	5	6	0†	7
A	0	0	0	2	1

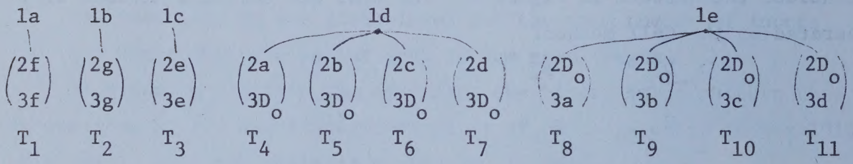
"†" - D_0 test pattern

"×" - D_1 test pattern

All lines being tested have no superscripts

#2	a	b	c	d	e	f	g	D_o
D	1	2	4	8	$15^{\times}$	0	$15^{\times}$	0†
E	1	2	4	8	$15^{\times}$	$15^{\times}$	0	0†
F	1	2	4	8	0	$15^{\times}$	$15^{\times}$	0†
B	7	7	7	7	6	3	5	0†

#3	a	b	c	d	e	f	g	D_o
G	1	2	4	8	$15^{\times}$	0	$15^{\times}$	0†
H	1	2	4	8	$15^{\times}$	$15^{\times}$	0	0†
I	1	2	4	8	0	$15^{\times}$	$15^{\times}$	0†
C	7	7	7	7	6	3	5	0†


1a
1b
1c
1d
1e
2f 3f
2g 3g
2e 3e
2a 3D_o
2b 3D_o
2c 3D_o
2d 3D_o
2D_o 3a
2D_o 3b
2D_o 3c
2D_o 3d
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11

The eleven tests are therefore:

	D	E	F	G	H	I
T_1	0	$15^{\times}$	$15^{\times}$	0	$15^{\times}$	$15^{\times}$
T_2	$15^{\times}$	0	$15^{\times}$	$15^{\times}$	0	$15^{\times}$
T_3	$15^{\times}$	$15^{\times}$	0	$15^{\times}$	$15^{\times}$	0
T_4	1	1	1	0†	0†	0†
T_5	2	2	2	0†	0†	0†
T_6	4	4	4	0†	0†	0†
T_7	8	8	8	0†	0†	0†
T_8	0†	0†	0†	1	1	1
T_9	0†	0†	0†	2	2	2
T_{10}	0†	0†	0†	4	4	4
T_{11}	0†	0†	0†	8	8	8

Theorem 1: N tests detect all s-a-faults in a regular tree network where $N = n_1 n_3 n_5 \ldots + n_2 n_4 n_6 \ldots$ (n_i is the number of inputs to each gate in level i).

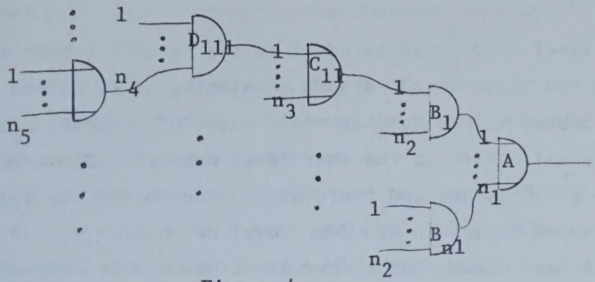


Figure 4

Proof: Gate A in Figure 4 is completely tested by $n_1 + 1$ tests. One test ($A = 00...0$) checks all inputs simultaneously for s-a-1 faults while the remaining n_1 tests are needed to check each of the n_1 inputs individually for s-a-0 faults (while all other lines are desensitized).

Assume a faulty "1" appears on an input line of Gate A while all inputs are simultaneously tested for s-a-1 faults. The fault is detected at the output unless the output is s-a-0 but a fault at the output is detected.

Assume a faulty "0" appears on an input line of Gate A while it is being individually tested for a s-a-0 fault. The fault is detected at the output unless it is masked by a faulty "1" on one of the lines not being tested. Since the lines not being tested are desensitized to "0", they have a faulty "1" only if they are s-a-1. This implies the output is s-a-1, but a fault at the output is detected.

Each Gate B_i ($i = 1,2, \ldots n_1$) requires $n_2 + 1$ tests. One test checks all inputs simultaneously for s-a-0 faults while the remaining n_2 tests are needed to check each of the n_2 inputs individually for s-a-1 faults (while all other lines are desensitized).

All B_i gates are simultaneously tested for "0" outputs (there are n_2 such tests). Each gate B_i is individually tested for "1" output (there are n_1 such tests). Since faults appearing at the inputs of Gate A are detected, this implies that $n_1 + n_2$ tests detect any faults in the two level network.

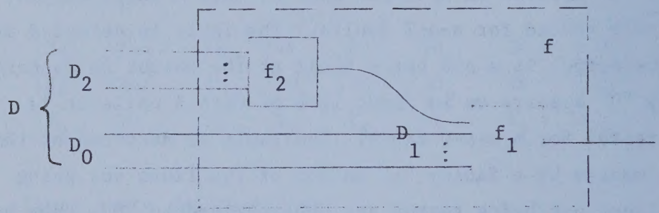
By similar reasoning $n_2 + n_3$ tests are needed to detect any faults in the two level network with Gate B_i as the output. n_2 of these tests have "0" outputs (on Gate B_i). All B_i gates are simultaneously tested for "0" outputs, therefore n_2 tests suffice to detect s-a-1 faults in the three level network consisting of all Gates A, B, and C. Each B_i gate is individually tested for "1" output, there are n_1 such gates each of which requires n_3 tests, therefore $n_1 n_3$ tests suffice to detect s-a-0 faults in the three level network. The three level network is thus completely tested by $n_1 n_3 +$

n_2 tests.

If a level of D gates is added to the network (where each D gate has n_4 inputs) the three level network consisting of B, C, and D gates requires $n_2n_4 + n_3$ tests. Since n_2n_4 of these have "0" outputs, n_2n_4 tests suffice to detect s-a-1 faults in the four level network. Since there are n_3 tests which have a "1" output and there are n_1 such networks, n_1n_3 tests suffice to detect s-a-0 faults in the four level network.

It has been shown that a four level network is completely tested by $n_1n_3 + n_2n_4$ tests. The above procedure may be repeated to generate the test set for the general n level network.

Theorem 2: In the following network let D_1, D_2 and D be inputs to networks f_1, f_2, and f respectively. D is the desensitizing test for f iff D_1 and D_2 are desensitizing tests for f_1 and f_2 respectively.



$$D = D_2D_0$$
$$D_1 = f_2(D_2)D_0$$
$$f(D) = f_1[f_2(D_2)D_o] = f_1(D_1)$$

Figure 5

Proof: The proof of this theorem will appear in [8].

Lemma 1: If $D_a = x_1x_2 \dots x_n$ is the desensitizing "a" test (where a=0 or 1), then $D_{\bar{a}} = \bar{x}_1\bar{x}_2 \dots \bar{x}_n$.

Proof: The proof of this theorem will appear in [8].

MULTIPLE FAULT TEST GENERATION PROCEDURE FOR NETWORKS WITH RECONVERGENT FANOUT

A) Reconvergence Limited To Within Individual Cells.

The Cell Method must be modified if reconvergent fanout exists in the network. The modified procedure follows:

1) Modify the network into the alternating AND/OR form wherever possible.
2) Break the network down into the overlapping cells (overlap between cells is one gate, also reconvergence is limited to within individual cells.)
3) Generate the complete test set for each cell. Check to see that all the necessary inputs to the gates are possible. If not make appropriate modifications.

4) When lines which fanout require simultaneous testing, do the following:

Keep all of these lines, except one, fixed to a given input pattern and vary the input patterns to the line being tested. Repeat the process until all the lines in question have been tested.

NOTE: Once a line has been tested for a given input pattern it does not need to be tested again for this same pattern. If a pattern must be repeated for another test, mark it to denote that this pattern does not need to be tested again.

5) Expand on the partial test set in the same way as the original Cell Method.

The procedure is best illustrated by an example.

EXAMPLE

The test set for the network in Figure 6 will be generated by the method described above.

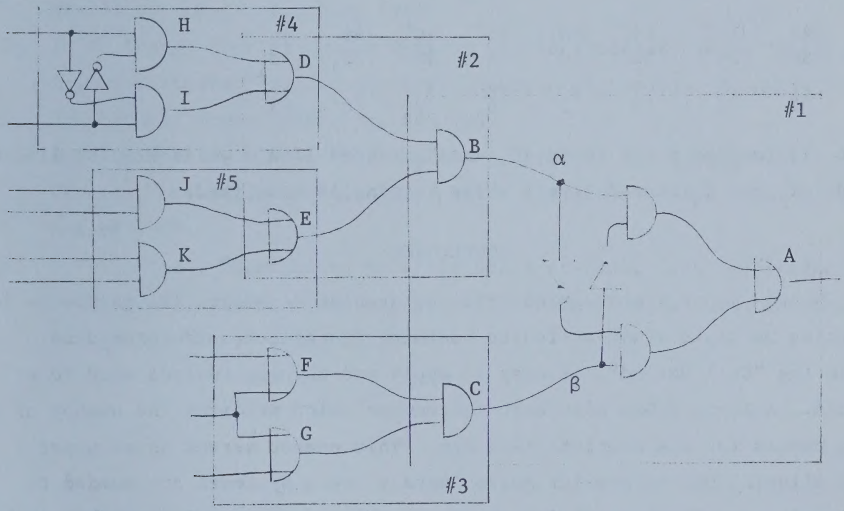


Figure 6

#1	a	b	c	d	e
B	3	3	3	2	1
C	3	2	1	3	1
A	0	2	2	1	0

#2	a	b	c	d
D	1	2	~~3~~ 2	0
E	1	2	0	3
B	3	3	2	1

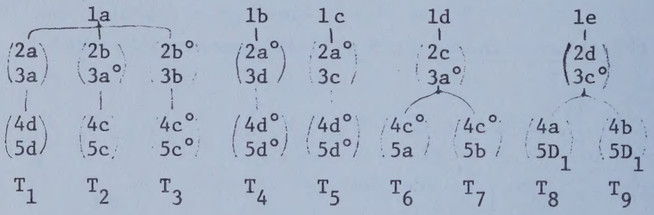
#3	a	b	c	d
F	2	1	0	1
G	1	2	1	0
C	3	3	1	2

D = 3 is not possible ∴ modify it to D = 2 (or D = 1)

#4	a	b	c	d
H	2	1	3	0
I	1	2	0	3
D	0	0	2	1

#5	a	b	c	d	D_1
J	1	2	3	0	3
K	1	2	0	1	3
E	0	0	2	1	3

This test was added since E = 3 was needed.


1a
1b
1c
1d
1e
2a 3a
2b 3a°
2b° 3b
2a° 3d
2a° 3c
2c 3a°
2d 3c°
4d 5d
4c 5c
4c° 5c°
4d° 5d°
4d° 5d°
4c° 5a
4c° 5b
4a 5D1
4b 5D1
T1
T2
T3
T4
T5
T6
T7
T8
T9

"o" - This pattern is not being tested

NOTE: 1a implied three tests, T_1 and T_2 tested line α while keeping line β fixed. T_2 and T_3 tested line β while keeping line α fixed.

CONCLUSION

In this paper a new approach to the problem of generating test sets for detecting multiple stuck-at-faults has been introduced. The procedure, called the "Cell Method", is easy to apply and reduces tedious work to a minimum. A formula has also been introduced which predicts the number of tests needed for the complete test set. This number serves as an upper bound allowing one to predict approximately how many tests are needed to completely test the network in question. This prediction is possible without actually having to generate the tests.

REFERENCES

[1] J. W. Gault, J. P. Robinson, and S. M. Reddy, "Multiple Fault Detection in Combinational Networks," IEEE Trans. Comput., Vol. C-21, pp. 31-36, January 1972.

[2] A. D. Friedman, "Fault Detection in Redundant Circuits," IEEE Trans. Electron. Comput. (Short Notes), Vol. EC-16, pp. 99-100, February 1967.

[3] D. R. Schertz, and G. A. Metze, "On the Indistinguishability of Faults in Digital Systems," in Proc. 6th Annual Allerton Conference Circuit and System Theory, 1968, pp. 752-760.

[4] D. C. Bossen, and S. J. Hong, "Cause-Effect Analysis for Multiple Fault Detection in Combinational Networks," IEEE Trans. Computers, Vol. C-20, pp. 1252-1257, November 1971.

[5] D. B. Armstrong, "On Finding a Nearly Minimal Set of Fault Detection Tests for Combinational Networks," IEEE Trans. Electron. Comput., Vol. ED-15, pp. 66-73, February 1966.

[6] J. F. Poage, "Derivation of Optimum Tests to Detect Faults in Combinational Circuits," in Mathematical Theory of Automata. Brooklyn, NY; Polytechnic Press, 1963, pp. 483-528.

[7] G. J. Lipovski, "An Improved Method of Finding All Largest Combinable Classes," Coordinated Sci. Lab., Univ. Illinois, Urbana, Rep. R-362, August 1967.

[8] C. D. Latino, "Simplified Multiple Stuck-At-Fault Test Generation Techniques," Doctoral Dissertation, The Pennsylvania State University, University Park, to be published.

ON THE EXISTENCE OF COMBINATIONAL NETWORKS WITH ARBITRARY MULTIPLE REDUNDANCIES

JAMES E. SMITH and GERNOT METZE
University of Illinois, Urbana, Illinois

ABSTRACT

Redundancy involving a single line is fairly well understood. However, little is known about multiple-line redundancies for which any proper subset is irredundant. This paper provides some tools for studying such redundancies. It presents several new examples of multiple redundancies and proves by a constructive method that redundancies of all multiplicities exist.

1. INTRODUCTION

A combinational logic network is said to be redundant if certain gate input or output lines may be removed without affecting the function realized by the network. Removing a line is equivalent to setting it to a constant logical value which depends on the type of gate fed by the line; e.g. removing an input line to an AND or NAND gate is equivalent to setting that input to logical 1. Redundancy is useful in some applications to increase system reliability [1], however, unintentional redundancy may increase the cost of a network without yielding specific benefits; in fact, redundancy may make it difficult or impossible to diagnose all faults in a network.

Many hardware failures in logic networks can be conveniently modelled as gate inputs or outputs stuck-at-0 or stuck-at-1. A fault is diagnosed by applying certain input vectors selected such that the fault-free and the faulty network produce different outputs. It should be evident that it is impossible to detect whether, for example, a redundant input line to an AND gate has become stuck-at-1. Thus, a redundant line corresponds to an undetectable stuck-line fault. Since the occurrence of undetectable faults may interfere with the diagnosis of otherwise detectable faults [2], methods for avoiding, or at least identifying, unintentional redundancy are needed.

Redundancy which involves a single line is fairly well understood and practical methods for identifying single-line redundancies have been proposed [3]. However, there exist networks having a set of redundant lines for which any proper subset is irredundant. Little is known about such multiple redundancies; in fact, only one example of a network containing a 4-tuple redundancy has been (repeatedly) published [4].

This paper, which is based on a more detailed report [8], provides some tools for further studies of multiple redundancies. We present several examples which suggest ways in which multiple redundancies occur and present a method for constructing circuits with redundancies of any multiplicity. This paper makes it apparent that multiple redundancy is not as rare as has been suspected and could easily occur in practical situations.

This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under contract DAAB-07-72-C-0259.

2. DEFINITIONS AND NOTATION

It is often more convenient to discuss redundancy in terms of undetectable faults rather than removable lines. A fault F in a network N consists of one or more stuck-line components $F = \{\ell_1/w_1, \ell_2/w_2, \ldots, \ell_k/w_k\}$ where ℓ_i/w_i denotes the fault "line ℓ_i stuck-at-w_i." The multiplicity of the fault is equal to the number of its component faults. A fault F is undetectable if and only if, for all possible input vectors x_t, the output of the network with fault F is the same as the output of the fault-free network. A network contains a k-line redundancy if there exists a fault F of multiplicity k that is undetectable but any proper subset of F is detectable. A network is k-redundant if it contains a k-line redundancy but no j-line redundancy for $j<k$.

The network N used here is a single-output combinational network having the internal structure shown in Figure 1: The network function Z(X) is realized by subnetwork N_o as the function Z(Y) of the intermediate function vector $Y = (y_1, y_2, \ldots, y_j)$ whose components in turn are realized by subnetworks $N_1, N_2, \ldots, N_j$ as functions of the primary inputs $X = (x_1, x_2, \ldots, x_m)$, i.e. $Z(X) = Z(Y(X))$.

A necessary and sufficient condition for the undetectability of a multiple fault $F = (f_1, f_2, \ldots, f_k)$ in a network N will now be proved. Let Y^F and Z^F denote the functions realized by the respective subnetworks under fault F. Define two sets, S_F and T_F, of ordered pairs (Y, Y^F) of intermediate function vectors as follows:

$$S_F = \{(Y(X_a), Y^F(X_a)) \quad \text{for every } X_a\}$$

$$T_F = \{(Y(X_a), Y^F(X_b)) \mid Z(Y(X_a)) = Z^F(Y^F(X_b)) \text{ for any } X_a, X_b\}.$$

Theorem 1: A multiple fault F in a network N is undetectable iff $S_F \subseteq T_F$.

Proof: Let (A,B) denote any ordered pair belonging to S_F and let {X} denote the set of input vectors for which $Y(X) = A$ and $Y^F(X) = B$. By definition, {X} is non-empty. It must be shown that, for any $(A,B) \in S_F$, F is undetectable iff $(A,B) \in T_F$.

If $(A,B) \in T_F$, then $Z(A) = Z^F(B)$ or $Z(Y(X)) = Z^F(Y^F(X))$ for all $X \in \{X\}$, i.e. the fault-free output is the same as the output under fault F, for every input vector $X \in \{X\}$. Therefore, F is undetectable if $\{(A,B) \in S_F\} \subseteq T_F$.

If $(A,B) \notin T_F$, then $Z(A) \neq Z^F(B)$ and $Z(Y(X)) \neq Z^F(Y^F(X))$ for all $X \in \{X\}$, i.e. each input vector X constitutes a test that detects F. By contraposition, if F is undetectable then $\{(A,B) \in S_F\} \subseteq T_F$. Q.E.D.

Example 1: Let $j=2$, $y_1 = x_1 x_2$, $y_2 = x_1 x_2 x_3 \vee \bar{x}_1 \bar{x}_2$, and $Z = y_1 \oplus y_2 = \bar{y}_1 y_2 \vee y_1 \bar{y}_2$. This network is shown in Figure 2.

Consider the multiple fault $F = \{\alpha/1, \beta/1\}$ shown in the figure. Then

$$y_1^F = x_2, \; y_2^F = x_1 x_2 x_3 \vee \bar{x}_1, \text{ and } Z^F = \bar{y}_1 y_2 \vee y_1 \bar{y}_2 = Z.$$

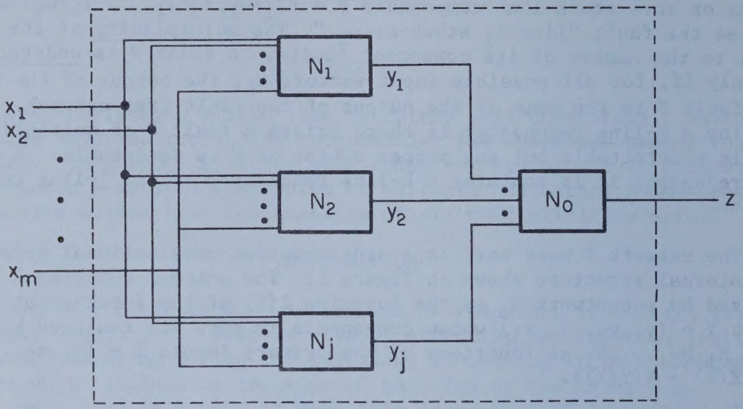

x_1
x_2
x_m
N_1
N_2
N_j
N_0
y_1
y_2
y_j
z

Figure 1. Structure of Network N

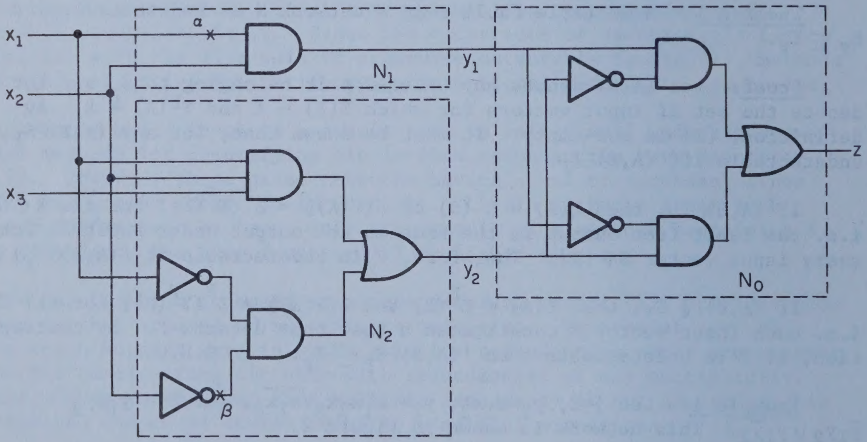

x_1
x_2
x_3
α
β
N_1
N_2
N_0
y_1
y_2
z

Figure 2. Network for Example 1

It is evident that

$$(Y,Y^F) = (01,01) \text{ for } \{X\} = \{000,001\},$$

$$(Y,Y^F) = (00,11) \text{ for } \{X\} = \{010,011\}, \text{ etc.},$$

and hence

$$S_F = \{(01,01),(00,11),(00,00),(10,10),(11,11)\}.$$

Further,

$$Z = 1 \text{ for } \{Y\} = \{01,10\}, \quad Z^F = 1 \text{ for } \{Y^F\} = \{01,10\},$$

and $Z = 0$ for $\{Y\} = \{00,11\}$, $Z^F = 0$ for $\{Y^F\} = \{00,11\}$;

hence

$$T_F = \{(01,01),(01,10),(10,01),(10,10),(00,00),(00,11),(11,00),(11,11)\}.$$

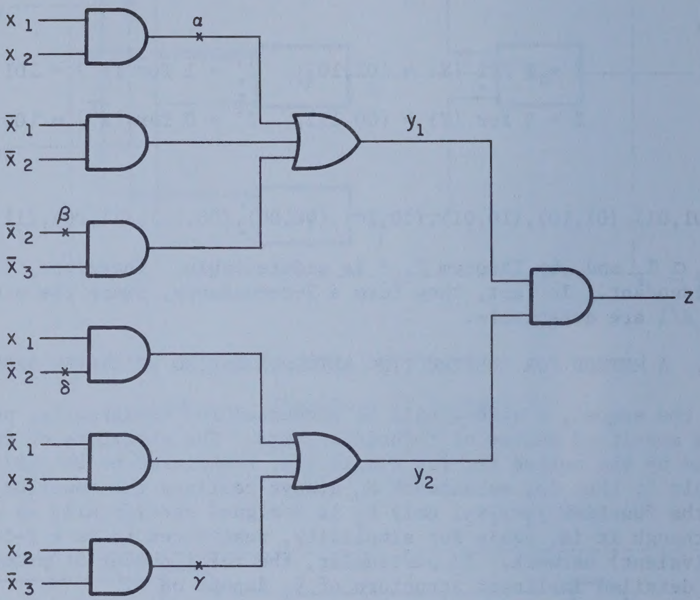
Thus, $S_F \subseteq T_F$ and, by Theorem 1, F is undetectable. Therefore, lines α and β are redundant. In fact, they form a 2-redundancy, since the single faults $\alpha/1$ and $\beta/1$ are detectable.

3. A METHOD FOR CONSTRUCTING NETWORKS HAVING SPECIFIED REDUNDANCIES

In the sequel, a method will be presented for constructing networks having a specified degree of redundancy, $k \geq 3$. The structure of the networks generated by the method is, for simplicity, restricted to be similar to that of Example 1, that is, subnetwork N_o always realizes the function $Z=y_1 \oplus y_2$ and N_1 the function $y_1=x_1x_2$, only N_2 is designed specifically as a function of k although it is, again for simplicity, restricted to be a 2-level AND-OR (or equivalent) network. In particular, the total number of primary inputs and the detailed implicant structure of y_2 depend on k.

There undoubtedly exist many other network structures which could be made to exhibit arbitrary redundancies. For example, the network of Figure 3 contains the (non-unique) 4-line redundancy shown. However, the subnetwork structure of Figure 2 is sufficient and convenient for our purposes since the functional behavior under a fault F is easily determined. In particular, each AND-gate in a two-level AND-OR subnetwork realizes an implicant of the subnetwork function, and stuck-line faults in the subnetwork are equivalent to the growth or disappearance of these implicants. Undetectable faults, and hence redundancies, will occur unless the AND-OR network realizes an irredundant cover of prime implicants [5].

The desired k-redundancy occurs in our networks because a single fault in N_1 and a (k-1)-tuple fault in N_2 mask each other; either of these faults is detectable by itself and, in fact, all three subnetworks are irredundant with respect to their outputs. It will be shown later that our procedure yields an irredundant cover of prime implicants for y_2 and that all faults of multiplicity less than k occurring anywhere in the entire network are detectable.


x 1
x 2
α
x̄ 1
x̄ 2
y1
β
x̄ 2
x̄ 3
z
x 1
x̄ 2
δ
x̄ 1
x 3
y2
x 2
x̄ 3
γ

α,γ: stuck-at-0 faults

β,δ: stuck-at-1 faults

x_1	x_2	x_3	y_1	y_2	Z	y_1^F	y_2^F	Z^F
0	0	0	1	0	0	1	0	0
0	0	1	1	1	1	1	1	1
0	1	0	0	1	0	1	0	0
0	1	1	0	1	0	0	1	0
1	0	0	1	1	1	1	1	1
1	0	1	0	1	0	0	1	0
1	1	0	1	1	1	1	1	1
1	1	1	1	0	0	0	1	0

$$S_F = \{(01,01),(01,10),(10,10),(11,11)\}$$

$$\subseteq T_F = \{(00 \text{ or } 01 \text{ or } 10,\ 00 \text{ or } 01 \text{ or } 10),(11,11)\}$$

Figure 3. A 4-redundant Network

Procedure for Generating the Implicants of y_2

Let

k = desired degree of redundancy ≥ 3
$c = [\log_2 k]$, thus $2^{c-1}+1 \leq k \leq 2^c$
n = total number of input variables
$= c+2$.

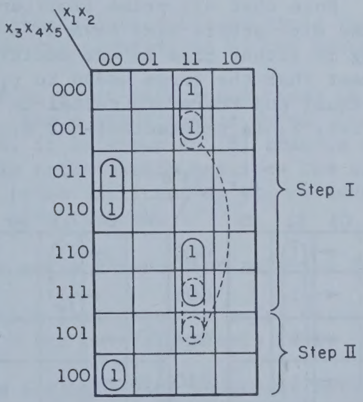
Construct a map in which the two variables x_1x_2 identify the columns and the remaining c variables identify the rows, in the standard reflected binary code. Thus there will be 4 columns and 2^c rows. The procedure specifies the minterms of y_2 by placing a 1 into each of the 2^c rows, either in column 11 or in column 00.

Step I: Form 2-loops alternately in column 11 and in column 00 by placing pairs of 1's two rows at a time, until 2^c+1-k 2-loops are formed. (This number of implicants ranges from 1 when $k=2^c$ to 2^{c-1} when $k=2^{c-1}+1$.)

Step II: Consider the remaining $2(k-2^{c-1}-1)$ rows one at a time. Alternate placing a 1 into column 11 and into column 00. Combine each minterm with an adjacent one (from Step I) into a 2-loop whenever possible. A total of $2(k-2^{c-1}-1)$ implicants is formed in this step. (This number ranges from 2^{c-2} when $k=2^c$ to 0 when $k=2^{c-1}+1$.)

The total number of implicants generated by the Procedure is $(2^c+1-k) + 2(k-2^{c-1}-1)=k-1$. The following example illustrates the process:

Example 2: Let $k=6$. Then $c=3$ and $n=5$. There are $2^c+1-k = 8+1-6 = 3$ implicants created in Step I and $2(k-2^{c-1}-1) = 2(6-4-1) = 2$ implicants in Step II. The resulting map for y_2 is shown below.



$$y_2 = x_1x_2\overline{x}_3\overline{x}_4 \vee \overline{x}_1\overline{x}_2\overline{x}_3x_4 \vee x_1x_2x_3x_4 \vee x_1x_2\begin{bmatrix} x_4 \\ x_3 \end{bmatrix}x_5 \vee \overline{x}_1\overline{x}_2x_3\overline{x}_4\overline{x}_5 \qquad \square$$

The Procedure specifies the minterms of y_2 by placing a 1 into each of the 2^c rows of the map, either in column 11 or in column 00. Thus, each minterm is of the form $x_1x_2x_3^*x_4^* \ldots x_n^*$ or $\overline{x}_1\overline{x}_2x_3^*x_4^* \ldots x_n^*$, where x_1x_2 or

$\overline{x}_1\overline{x}_2$ is the column identifier or "prefix" and $x_3^*x_4^* \ldots x_n^*$ is the row identifier, with x_i^* meaning x_i or $\overline{x}_i$. Let q_j denote a row identifier $x_3^*x_4^* \ldots x_n^*$ and let Q^{11} denote the set of row identifiers with prefix x_1x_2 and Q^{00} the set of row identifiers with prefix $\overline{x}_1\overline{x}_2$. Note that each row identifier q_j appears exactly once, either in Q^{11} or in Q^{00}; thus $Q^{11} \vee Q^{00} = 1$.

Lemma 1: The implicants of y_2 generated by the procedure from an irredundant, minimal cover of y_2. The proof of Lemma 1 is rather lengthy and is not given here; however, a proof appears in [8].

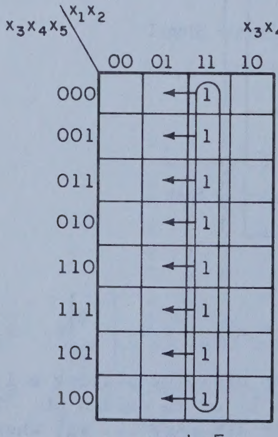
If subnetwork N_2 is constructed to realize y_2 in two-level AND-OR (or equivalent) form, with k-1 AND gates corresponding to the k-1 prime implicants of y_2, then N_2 will be single- and multiple-line irredundant [5]. Of course, N_1 and N_0 are also single- and multiple-line irredundant by inspection.

It now remains to be shown that the composition of subnetworks N_0, N_1, and N_2 in the form of Figure 2 contains at least one k-line redundancy but no redundancy of lower cardinality.

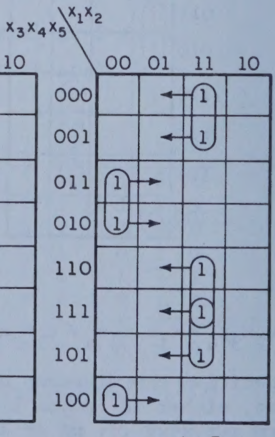
4. PROOF OF k-REDUNDANCY

We claim that the following k-tuple fault F in N is undetectable, but all faults of lower multiplicity are detectable: F consists of subfaults F_1 and F_2 where F_1 is the stuck-at-1 fault at the x_1 input to the AND gate of N_1 and F_2 is the (k-1)-tuple fault consisting of stuck-at-1 faults at either the x_1 or the $\overline{x}_2$ input to each AND gate of N_2, depending on whether the AND gate realizes a prime implicant with prefix x_1x_2 or $\overline{x}_1\overline{x}_2$.

Example 2, cont'd: Maps for y_1 and y_2 of Example 2 are shown below; the arrows indicate the growths of the prime implicants of y_1 and y_2 under F_1 and F_2, respectively. Note that all prime implicants of y_2 grow into the same column, because those with prefix x_1x_2 have $x_1/1$ and those with prefix $\overline{x}_1\overline{x}_2$ have $\overline{x}_2/1$, resulting in either case in the addition of terms with prefix $\overline{x}_1x_2$. It is evident that the terms added to y_1 by F_1 and those added to y_2 by F_2 are exactly equal and therefore cancel in N_0, since F has no component in N_0. Thus, $F=F_1 \cdot F_2$ is undetectable at Z. In the notation of



y_1 under F_1



y_2 under F_2

Theorem 1, (00,11) is the only member of S_F that is not in S_F under fault-free conditions, but since

$$T_F = \{(y_1y_2, y_1^Fy_2^F) \mid y_1^Fy_2^F = y_1y_2 \text{ or } \overline{y}_1\overline{y}_2\},$$

$(00,11) \in T_F$ and F is undetectable. □

It should be noted that the undetectable k-tuple fault F described above is not unique; due to the symmetry of the functions with respect to x_1 and x_2 an equally undetectable k-tuple fault F' consisting of stuck-at-1 faults at the x_2 and $\overline{x}_1$ inputs to the AND gates of N_1 and N_2 also exists.

Lemma 2: The networks constructed by the procedure described above for a given k are k-redundant, i.e. the k-tuple fault F is undetectable but any proper subfault of F is detectable.

Proof: (i) The undetectability of F was demonstrated in Example 2 above; a formal proof is omitted here.

(ii) We now show that any proper subfault $\hat{F}$ of $F = F_1 \cdot F_2$ is detectable. Since F_1 is a single-line fault and therefore has no proper subfaults, only four cases need be considered: $\hat{F} = F_1$, $\hat{F} = F_2$, $\hat{F} = \hat{F}_2$ where $\hat{F}_2$ is a proper subfault of F_2, and $\hat{F} = F_1 \cdot \hat{F}_2$. The first three cases correspond to the growth of one or more prime implicants in y_1 or y_2, but not in both, and can be detected easily by an appropriate test in the growth area. The case $\hat{F} = F_1 \cdot \hat{F}_2$ represents a fault that differs from F in that at least one component of F_2 is not included. Thus, at least one prime implicant of y_2 will not grow under $\hat{F}_2$ and will therefore fail to cancel part of the growth of y_1 under F_1; this can be detected by a test located in the growth area of F_2 but not of $\hat{F}_2$. Thus, all proper subfaults of F are detectable.
Q. E. D.

Lemma 3: There is no undetectable multiple fault F of multiplicity less than k in the networks constructed by the method given.

Proof: The proof is split into three cases on the basis of whether N_1 is fault-free ($y_1^F=y_1$), has a stuck output ($y_1^F=w$), or has a stuck input ($y_1^F=x_1$). In each case, it is shown in [8] that as a consequence $y_2^F=\overline{y}_2$ for some region; since this relationship requires the simultaneous growth and disappearance of some prime implicant of y_2, no such fault can exist. We are now ready for our main theorem.

Theorem 2: There exist k-redundant networks for all k≥1.

Proof: We will show that networks exist which contain an undetectable fault of multiplicity k but no undetectable fault of lower multiplicity.

1) k=1: This is a single-line redundant network. Many examples are known to exist.

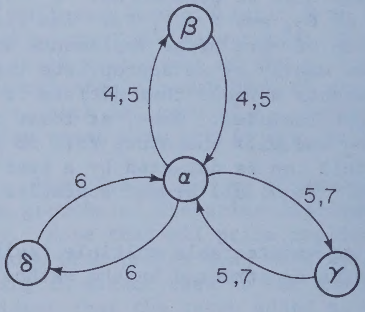
2) k=2: Consider the network of Example 1 which contains a 2-line redundancy and can easily be shown to be single-line irredundant. Thus, the network of Example 1 is 2-redundant.

3) k≥3: Networks generated by our Procedure satisfy the Theorem, as Lemmas 2 and 3 show.
Q.E.D.

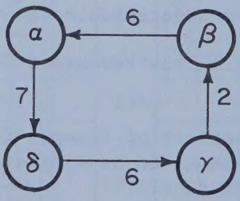
5. FAULT-MASKING IN REDUNDANT NETWORKS

A test input vector x_t designed to detect a fault F_o may fail to detect F_o in the presence of another fault. We say that a fault F_1 masks a fault F_o under test x_t if F_o but not $F_o \cdot F_1$ is detected by x_t. Fault masking relations may be expressed algebraically or graphically [6,7]. Whenever they form a fault masking cycle the set of faults involved in the cycle is undetectable by the test set; if all possible test sets exhibit fault masking cycles the fault set corresponds to a multiple-line redundancy.

We show in this section that several distinct patterns of fault masking cycles corresponding to multiple-line redundancies exist. For example, a 4-redundant network generated by our Procedure is shown in Figure 4 below, together with the undetectable 4-tuple fault $F=\{\alpha,\beta,\gamma,\delta\}$. The maps show that X=0100 or 0101 is a test for fault α or β but not $\alpha \cdot \beta$, X=0101 or 0111 is a test for fault α or γ but not $\alpha \cdot \gamma$, and X=0110 is a test for fault α or δ but not $\alpha \cdot \delta$. Thus, β masks α, and vice versa, under test 4 or 5 (=X in decimal). The masking graph for F, using $\alpha \xrightarrow{6} \delta$ to denote that fault α masks fault δ under test 6, is therefore



In contrast, the masking graph for the undetectable fault shown in the 4-redundant network of Figure 3 consists of a single loop,



as does the masking graph for the 4-tuple undetectable fault in [4].

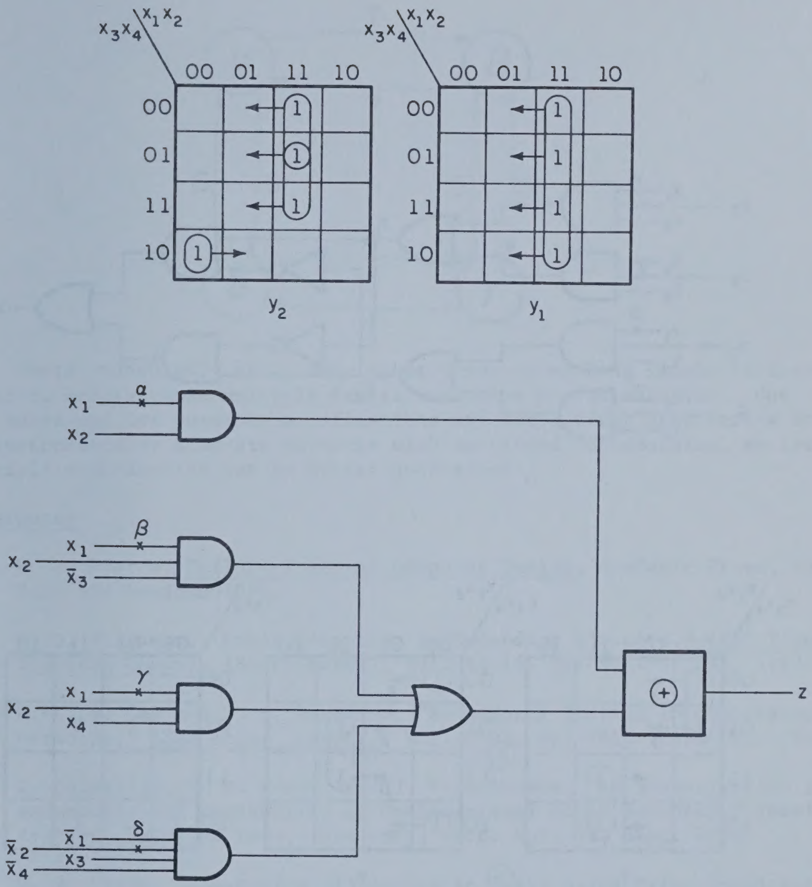

x1x2
x3x4
00
01
11
10
1
y2
y1
α
β
γ
δ
x1
x2
x̄3
x4
x̄1
x̄2
x3
x̄4
⊕
z

Figure 4. A 4-redundant Network
Generated by our Procedure

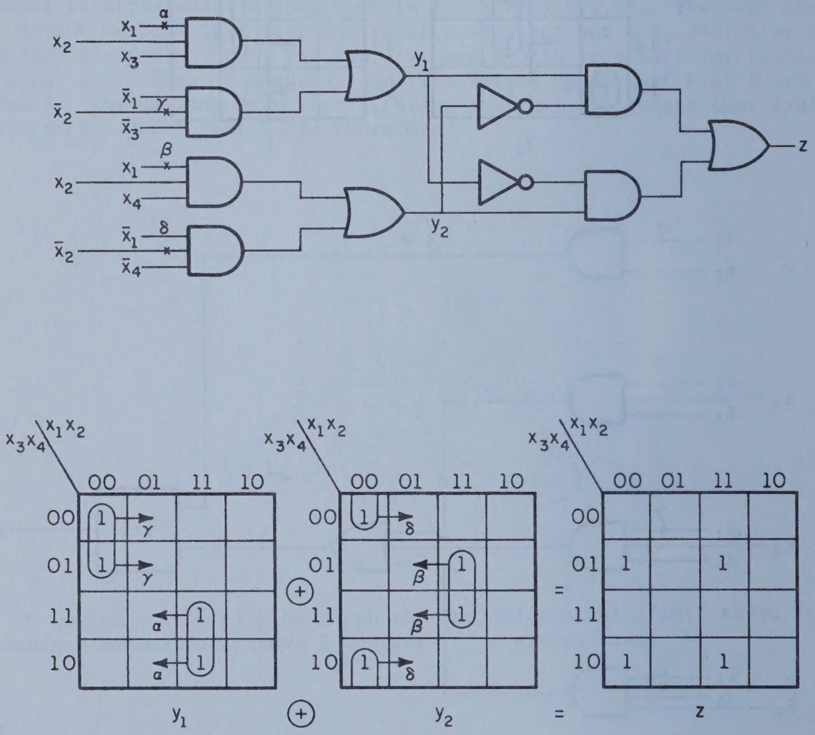
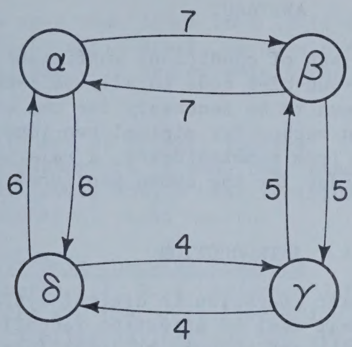

α
x̄1
x2
x3
x̄1
γ
x̄2
x̄3
β
x1
x2
x4
δ
x̄1
x̄2
x̄4
y1
y2
z
x3x4
x1x2
00
01
11
10
1
γ
α
y1
y2
z
δ
β
⊕
=

Figure 5. A 4-redundant Network Generated by a Modification of Our Procedure

Yet another type of masking graph can be obtained by modifying our Procedure to yield more complicated N_1 subnetworks. A 4-redundant network generated by the modified Procedure is shown in Figure 5. The masking graph for the undetectable 4-tuple fault $F=\{\alpha,\beta,\gamma,\delta\}$ consists of double loops:



There undoubtedly exist many other types of masking graphs that correspond to undetectable multiple faults and hence to redundancies. Our Procedure and its possible modifications are but a first step in the effort to systematically generate networks with specified redundancies, so that multiple redundancies can be better understood.

References

[1] W. H. Pierce, Failure-Tolerant Computer Design, Academic Press, New York and London, 1965.

[2] A. D. Friedman, "Fault Detection in Redundant Circuits," IEEE Trans. Electron Comput. (Short Notes), Vol. EC-16, pp. 99-100, Feb. 1967.

[3] H.-P. S. Lee and E. S. Davidson, "Redundancy Testing in Combinational Networks," IEEE Trans. Comput., Vol. C-23, pp. 1029-1047, Oct. 1974.

[4] R. Dandapain, S. M. Reddy, and J. P. Robinson, "An Investigation into Redundancy and Testability of Combinational Logic Networks," Themis Project, Univ. of Iowa, Iowa City, Tech. Rpt. 32, Sept. 1970.

[5] M. R. Paige, "Generation of Diagnostic Tests Using Prime Implicants," Report R-414, Coordinated Science Lab., University of Illinois, Urbana, May 1969.

[6] G. Metze and C. Cha, "Multiple Fault Diagnosis in Combinational Networks," Proc. Twelfth Annual Allerton Conf. on Circuit and System Theory, Monticello, Ill., Oct. 1974, pp. 244-253.

[7] F. J. O. Dias, "Fault Masking in Combinational Logic Circuits," Digest of Fourth Annual International Symp. on Fault-Tolerant Computing, June 1974, pp. 1-25 to 1-30.

[8] J. Smith and G. Metze, "On the Existence of Combinational Networks with Arbitrary Multiple Redundancies," Report R-692, Coordinated Science Lab., University of Illinois, Urbana.

GENERAL DESIGN RULES FOR THE CONSTRUCTION OF m-OUT-OF-n TOTALLY SELF-CHECKING CHECKERS

JAMES E. SMITH and GERNOT METZE
University of Illinois, Urbana Illinois

ABSTRACT

This paper presents a set of conditions sufficient to characterize a class of realizations of m-out-of-n code totally self-checking checkers. The same conditions are shown to be necessary for the construction of two-level checkers and a design method for minimal two-level checkers is given. Using a well known theorem from combinatorics, a relationship which must exist between m and n in order for the above mentioned conditions to be satisfied is shown.

1. INTRODUCTION

Achieving fault tolerant operation in digital systems through self-checking has received a great deal of attention recently [1,3,5,6]. In self-checking systems, inputs and outputs of logic function blocks, e.g. adders, are encoded in such a way that a fault within the block either does not affect the particular output of the block or transforms it into a invalid or noncode word. The validity of the output words is checked by checker circuits which produce "valid"/"invalid" indications and, of course, must check themselves during normal operation. That is, if a fault occurs within the check circuit itself, the same "invalid" indication is produced as for noncode inputs.

Some logic function blocks, e.g. memories, are particularly susceptible to unidirectional faults, i.e. faults whose components have all failed to the same value. Unidirectional faults are easily detected by m-out-of-n codes in which code words have precisely m 1's and n-m 0's and the change of any number of 1's to 0's, or vice versa, always results in a noncode word.

This paper presents a set of conditions which are sufficient to characterize a class of realizations of m-out-of-n code checkers that are self-checking for unidirectional faults. The conditions given greatly simplify the design of m-out-of-n code self-checking checkers. These same conditions are then shown to be necessary for the realization of m-out-of-n totally self-checking checkers in two levels, and the design of minimal two-level checkers is discussed.

For arbitrary m and n, the above mentioned conditions cannot always be met. Using a well known theorem from combinatorics, a relationship which m and n must satisfy in order for the conditions to be met is given.

Since many of the theorems presented here have rather lengthy proofs, the proofs have been omitted. However, the proofs to all the theorems appear in [15].

This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under contract DAAB-07-72-C-0259.

Representation and Modeling of Faults

The fault model used here is the usual stuck-at model [4]. That is, hardware failures in a logic circuit are modeled as some gate input or gate output lines stuck-at-1 (s-a-1) or stuck-at-0 (s-a-0). While the stuck-at fault model does not model all possible hardware failures, it models most of the more common failure modes and is easy to work with.

A fault occurs when some lines in a logic circuit become s-a-1 or s-a-0. When a single line is stuck, we have a single fault, and a circuit has a unidirectional fault if one or more lines are stuck at the same logic value.

The faults which are of primary interest in this report are unidirectional faults. The assumption is made that since the checker is being checked during normal operation, a unidirectional fault will be detected before a non-unidirectional fault occurs.

Properties of Totally Self-Checking Checkers

In his Ph.D. thesis [1], D. A. Anderson defined the concept of totally self-checking circuits and the properties which a totally self-checking checker must have. Similar definitions are given here.

Definition 1: A circuit is self-testing if, for every fault from a prescribed set, the circuit produces a noncode space output for at least one code space input.

Definition 2: A circuit is fault-secure if, for every fault from a prescribed set, the circuit never produces an incorrect code space output for code space inputs.

Definition 3: A circuit is totally self-checking for a prescribed fault set if it is both fault-secure for that set and self-testing for that set.

Definition 4: A circuit is code disjoint if the fault-free circuit always maps noncode space inputs to noncode space outputs.

Definition 5: A circuit is a totally self-checking checker if it is both totally self-checking and code disjoint.

For our checkers, the code space inputs are all m-out-of-n words, and our prescribed fault set is all unidirectional faults. Although the definition of totally self-checking checkers allows any number of outputs, it is customary to use the fewest necessary. It is easy to see that at least two are necessary [3], and in most cases two are sufficient. We will restrict ourselves to two outputs, as in [1], and let (0,1) and (1,0) be in the code space, and (1,1) and (0,0) in the noncode space. Based on the definition of totally self-checking checker, we now give the properties our checkers must have.

1) (Code disjoint property). The checker must map the $\binom{n}{m}$ code vertices onto the output pairs (0,1) or (1,0). The remaining (noncode) vertices are mapped onto either (0,0) or (1,1).

2) (Self-testing property). If a unidirectional fault occurs within the checker, at least one code word must be mapped onto (0,0) or (1,1).

3) (Fault-secure property). If a unidirectional fault occurs within the checker, no code word will be mapped onto the incorrect code output.

Notation

We consider an n input logic network to have the 2^n vertices of the n-cube as possible impu combinations. We write a particular vertex as

$$x = (x_1,x_2,\ldots,x_n) \quad \text{where} \quad x_1 \in \{0,1\}.$$

A vertex with exactly k ones will be called a k-vertex. Thus, the set of m-out-of-n code inputs will be the set of m-vertices of the n-cube. This set will also be called the set of code vertices at times.

We define the usual partial ordering on the vertices:

$$x \geq y \quad \text{iff} \quad x_i \geq y_i \quad \text{for} \quad i = 1,2,\ldots,n.$$

For example,

$$(1,1,1,1) \geq (1,1,0,1)$$

$$(1,1,0,1) \geq (1,0,0,1)$$

but,

$$(1,0,0,1) \not\geq (0,1,0,1).$$

We say that x covers y if $x \geq y$.

The complement of a vertex is formed by complementing all components. For example, the complement of (1,1,0,0) is (0,0,1,1).

Let the two logic functions which the two-output checker realizes be called f and g. This output pair will be written (f,g). For totally self-checking checkers, (f,g) = (0,1) or (1,0) for code input vertices and (f,g) = (0,0) or (1,1) for noncode input vertices.

2. THEOREMS CONCERNING THE CONSTRUCTION OF INVERTER-FREE TOTALLY SELF-CHECKING CHECKERS

This paper considers the class of totally self-checking checker networks which may be constructed using no inverters. Such inverter-free circuits have many desirable properties when unidirectional faults are being considered. All of the totally self-checking checker designs given previously [2,5] have used inverter-free networks.

Properties of Inverter Free m-out-of-n Totally Self-Checking Checkers

Two well known properties of inverter-free circuits will be given below as lemmas without proofs. The proofs follow directly from results presented by Betancourt [7].

Lemma 1: Let f be the function realized by an inverter-free logic network. Then, for any two input vertices x and y such that $x \geq y$, $f(x) \geq f(y)$.

If this lemma is applied to our checker output pair, we see that if $x \geq y$ then

$$(f(x),g(x)) \geq (f(y),g(y)).$$

Lemma 2: Let f be the function realized by an inverter-free logic network and let $f_0(f_1)$ be any function the network realizes under a unidirectional s-a-0 (s-a-1) fault. Then, for any input vertex x, $f_1(x) \geq f(x) \geq f_0(x)$.

If we let $(f,g)_0$ be the checker output pair under a s-a-0 unidirectional fault and let $(f,g)_1$ be the output pair under a s-a-1 unidirectional fault, then we see that for any input x, $(f(x),g(x))_1 \geq (f(x),g(x)) \geq (f(x),g(x))_0$.

Consider now an inverter-free but otherwise unconstrained realization of a totally self-checking m-out-of-n checker with outputs (f,g). Lemma 1 can be used to characterize the distribution of the noncode vertices into F and G, the sets of true vertices for f and g, as follows.

Theorem 1: In an inverter-free realization of a totally self-checking m-out-of-n checker with outputs f and g, all vertices with more than m 1's must be in both F and G, and no vertex with fewer than m 1's may be in either F or G.

Thus, the use of an inverter-free but otherwise unconstrained realization for totally self-checking m-out-of-n checkers completely determines which noncode vertices appear in F and G. Since Property 1 also requires that each code vertex belong either to F or to G but not to both, all that remains is the question of how to partition the code vertices into F and G.

We approach this question by constraining the realization of the checkers to be not only inverter-free but also irredundant and state two conditions on the partition of the code vertices into F and G.

Condition 1: Every (m+1)-vertex covers at least one m-vertex in F and at least one m-vertex in G.

Condition 2: Every (m-1)-vertex is covered by at least one m-vertex in F and at least one m-vertex in G.

These conditions are met by the partitions used in [1] as well as by the partition used in Example 1 below, although this is by no means easy to check. The conditions will be proved to be sufficient for any irredundant, inverter-free realization to be totally self-checking, and necessary for such realizations in two levels.

Sufficiency of Conditions 1 and 2 for a Totally Self-Checking Checker

Theorem 2: If Conditions 1 and 2 are met by the partition of the m-vertices of the n-cube into F and G, then any irredundant inverter-free network realizing f and g will be a totally self-checking m-out-of-n checker.

This theorem is proved in [15] independently of Theorem 1. Thus, the partition of the code vertices into F and G according to Conditions 1 and 2 is sufficient to determine the totally self-checking nature of the checker, and the distribution of the noncode vertices into F and G is implicitly determined.

Theorem 2 states that all of the code vertices are sufficient to test any realization of f and g. However, all of the code vertices are not always necessary as can be seen in Example 1.

Example 1: For a 3-out-of-6 checker, we may use as F and G:

F = {all vertices with more than m 1's and the following m-vertices:
(1,1,1,0,0,0),(1,1,0,1,0,0),(0,0,1,1,1,0),(0,0,1,1,0,1),
(1,0,0,0,1,1),(0,1,0,0,1,1)}

G = {all vertices with more than m 1's and all m-vertices not in F}.

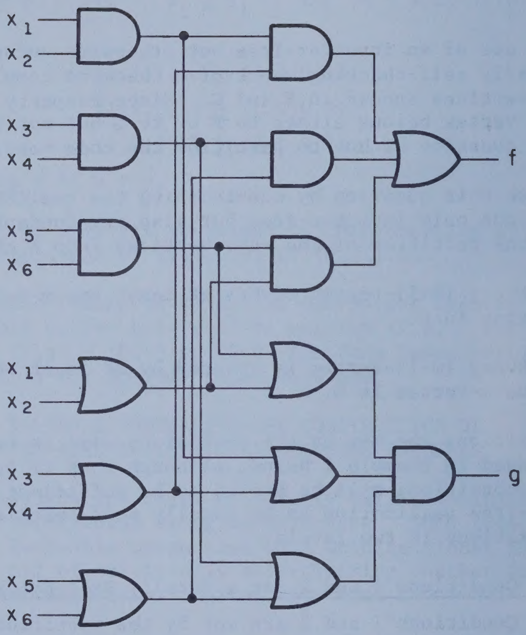

x 1
x 2
x 3
x 4
x 5
x 6
f
x 1
x 2
x 3
x 4
x 5
x 6
g

Figure 1. A Totally Self-checking 3-out-of-6 Checker

The above partition of code vertices into F and G is one of many which meets Conditions 1 and 2. One of the many realizations of f and g is shown in Figure 1. It can be verified that a complete test set for unidirectional faults in that circuit is $\{(0,1,1,1,0,0),(1,0,0,0,1,1),(1,1,0,1,0,0),(0,0,1,0,1,1),(1,1,0,0,0,1),(0,0,1,1,1,0)\}$, so that the set of code vertices is certainly sufficient.

Necessity of Conditions 1 and 2 for a Two-level Totally Self-Checking Checker

Since Conditions 1 and 2 are sufficient for any inverter-free realization to be totally self-checking, they are obviously sufficient for a two-level realization. As we will see in this section, they are also necessary for a two-level realization to be totally self-checking.

Lemma 3: In an irredundant two-level m-out-of-n totally self-checking checker, the only fanout which can occur is at the primary inputs.

As a consequence of Lemma 3, f and g are each either in AND-OR or OR-AND form with only primary inputs shared by the logic forming f and g.

Lemma 4: In an AND-OR relaization of f or g, each AND gate must have exactly m inputs. In an OR-AND realization of f or g, each OR gate must have exactly n-m inputs.

From the above lemma, we see that each AND gate in an AND-OR realization corresponds to one code vertex and has inputs corresponding to the coordinates of the vertex which are 1's. The AND gate produces the value 1 when the code vertex it corresponds to is present at the inputs and the value 0 when any other code vertex is present.

Each OR gate in an OR-AND realization also corresponds to one code vertex, and has inputs which correspond to the coordinates of the vertex which are 0's. The OR gate takes the value 0 when the code vertex it corresponds to is present at its inputs and the value 1 when any other code vertex is present.

Theorem 3: Conditions 1 and 2 are necessary for any two-level irredundant inverter-free m-out-of-n checker to be totally self-checking.

Minimal Two Level m-out-of-n Checkers

A totally self-checking system is composed of modules with information passing between the modules in encoded form. Each time information is passed, it must be checked before more information can be transferred. For this reason, it is important that the checking process be as fast as possible. Since the speed of a combinational circuit depends largely upon the number of logic levels used, it is easy to justify the desirability of two-level checkers. In this section, we will present a procedure for designing two-level totally self-checking checkers for m-out-of-n codes which use a minimum number of gates.

If a satisfactory partition of the m-vertices of an n-cube can be made, we let H(m,n) be the minimal number of code vertices that must be in block F. Then an AND-OR realization of f must have at least H(m,n)+1 gates since there is one AND gate for every m-vertex in F.

Similarly, an OR-AND realization of g must have at least H(m,n)+1 gates since one OR gate corresponds to each false m-vertex of g. Any other realizations of f or g under this partition can only require more gates. The same is true if a partition is used where $|\text{m-vertices in F}| > H(m,n)$ and $|\text{m-vertices in G}| > H(m,n)$. Therefore, a minimal two-level m-out-of-n checker must have at least 2H(m,n)+2 gates.

We can achieve this minimum by following the procedure just discussed. We should first partition the m-vertices according to Conditions 1 and 2 such that the number of m-vertices in F is minimal, then we should realize f using an AND-OR circuit and g using an OR-AND circuit. The inputs to the AND gates which realize f correspond to the 1 positions in the vertices in F, and the inputs to the OR gates which realize g correspond to 0 positions of the vertices in F. It is easy to see that such a realization performs the mapping specified by F and G and must be a totally self-checking checker.

Example 2: Minimal two-level 3-out-of-6 totally self-checking checker. Let F and G be the same sets as in Example 1. (For this partition $|\text{m-vertices in F}|$ is minimum.) The checker is shown in Figure 2.

A Lower Bound on H(m,n)

A nonexhaustive procedure for generating a partition such that $|\text{m-vertices in F}| = H(m,n)$ has not been found, and this partition is believed to be quite difficult in the general case. In fact the numbers H(m,n) are not known in general, although we can derive a lower bound on H(mn,) which appears to be rather tight for $m \simeq \left\lfloor \frac{n}{2} \right\rfloor$.

Let $H^-(m,n)$ be the minimum number of m-vertices of the n-cube which are required to cover all (m-1)-vertices, and let $H^+(m,n)$ be the minimum number of m vertices we must have in order for all (m+1)-vertices to cover at least one m-vertex. We observe that $H(m,n) \geq \max(H^-(m,n), H^+(m,n))$.

The covering of (m-1)-tuples by m-tuples has been treated rather extensively in combinatorics [13,14]. From [13] we get a bound on $H^-(m,n)$; in particular,

$$H^-(m,n) \geq \left\lceil \frac{n}{m} \left\lceil \frac{n-1}{m-1} \left\lceil \cdots \left\lceil \frac{n-m+3}{3} \left\lceil \frac{n-m+2}{2} \right\rceil \right\rceil \cdots \right\rceil \right\rceil \right\rceil .$$

Due to the same type of 'duality' we observe between Conditions 1 and 2 we see that

$$H^+(m,n) = H^-(n-m,n).$$

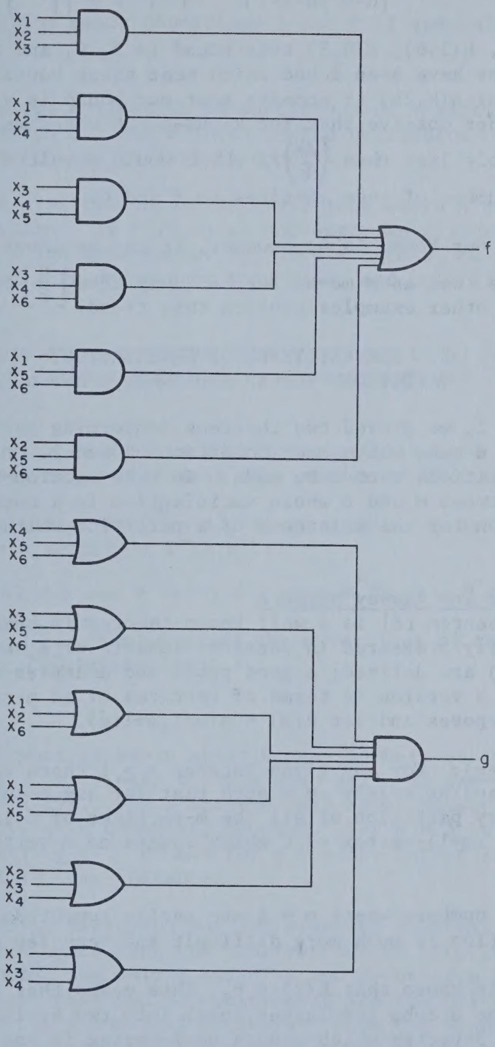

x_1
x_2
x_3
x_1
x_2
x_4
x_3
x_4
x_5
x_3
x_4
x_6
x_1
x_5
x_6
x_2
x_5
x_6
f
x_4
x_5
x_6
x_3
x_5
x_6
x_1
x_2
x_6
x_1
x_2
x_5
x_2
x_3
x_4
x_1
x_3
x_4
g

Figure 2. A Minimal Two-Level 3-out-of-6
Totally Self-checking Checker

Therefore, $H^{+}(m,n) \geq \left\lceil \frac{n}{n-m} \left\lceil \frac{n-1}{n-m-1} \left\lceil \cdots \left\lceil \frac{m+3}{3} \left\lceil \frac{m+2}{2} \right\rceil \right\rceil \cdots \right\rceil \right\rceil \right\rceil$, and

$$H(m,n) \geq \max\left(\left\lceil \frac{n}{m} \left\lceil \frac{n-1}{m-1} \left\lceil \cdots \left\lceil \frac{n-m+3}{3} \left\lceil \frac{n-m+2}{2} \right\rceil \right\rceil \cdots \right\rceil \right\rceil \right\rceil , \right.$$

$$\left. \left\lceil \frac{n}{n-m} \left\lceil \frac{n-1}{n-m-1} \left\lceil \cdots \left\lceil \frac{m+3}{3} \left\lceil \frac{m+2}{2} \right\rceil \right\rceil \cdots \right\rceil \right\rceil \right\rceil \right).$$

For $H(2,4)$, $H(3,6)$, $H(4,8)$ this bound is 2, 6, and 14 respectively. Actual partitions have been found which meet these bounds exactly in these cases. Thus, for $H(k,2k)$ it appears that our bound is very close to being exact. We further observe that for k-out-of-2k codes we can expect $H(k,2k)$ to be considerably less than $\binom{2k}{k}/2$ which would result from a partition with an equal number of code vertices in F and G.

For $H(2,5)$ our bound is 4; however, it can be shown that $H(2,5) = 5$. This is evidence that as m moves further away from $\left\lfloor \frac{n}{2} \right\rfloor$ our bound becomes less exact, and other examples confirm this trend.

3. THE EXISTENCE OF SATISFACTORY PARTITIONS FOR AN ARBITRARY m AND n

In Section 2, we proved two theorems concerning partitions of the m-vertices of the n-cube which meet Conditions 1 and 2. However, for some m and n, such partitions cannot be made. In this section we will present a relationship between m and n whose satisfaction is a necessary and sufficient condition for the existence of a partition which meets Conditions 1 and 2.

Ramsey's Theorem and Ramsey Numbers

Ramsey's theorem [8] is a well known theorem in combinatorics. The theorem is usually presented in terms of subsets of a set, and the Ramsey numbers $N(p,q;r)$ are defined; a good proof and examples can be found in [9]. We present here a version in terms of vertices of an n-cube that is specialized for our purposes and let $R(m) = N(m+1,m+1;m)$.

Theorem 4 (Ramsey): For any given integer $m \geq 1$ there exists a finite number $R(m)$ depending solely on m such that for any n-cube C, $n \geq R(m)$, and for any arbitrary partition of all the m-vertices of C into two blocks, there exists an (m+1)-vertex of C which covers no m-vertex in one of the blocks.

The Ramsey numbers where $m = 1$ are easily computed, however if $m \geq 2$ their determination is much more difficult and very few are known [9].

Example 3: It is known that $R(2) = 6$. This means that if we partition the 2-vertices of the 6-cube (or larger cube) into two blocks in any manner, there must be a 3-vertex which covers no 2-vertex in one block of the partition. Conversely since $R(2)$ is the least number having these properties, at least one 2-block partition of the 2-vertices of a 5-cube (or smaller cube) exists such that every 3-vertex covers at least one 2-vertex in each block, as in the partition

$$A = \{12,14,25,34,35\}, \ B = \{13,15,23,24,45\}.$$

We see that Conditions 1 and 2 are in some sense duals. That is, if we can get a partition for an m-out-of-n code, we can also get a partition for an (n-m)-out-of-n code by simply complementing the partitioned m-vertices. For this reason, without loss of generality, we will consider only the case where $m \leq \left\lfloor \frac{n}{2} \right\rfloor$.

We will show that we can partition the m-out-of-n vertices, $m \leq \left\lfloor \frac{n}{2} \right\rfloor$, such that the partition meets Conditions 1 and 2 if and only if $R(m) > n$.

Main Theorem

Let $P^{+}(m,n)$ be the collection of partitions of m-vertices of the n-cube which satisfy Condition 1. Let $P^{-}(m,n)$ be the collection of partitions of m-vertices of the n-cube which satisfy Condition 2.

Let $A|B$ denote a partition of the m-vertices where A and B are the two blocks of the partition. If $P^{-}(m,n)$ is not empty an arbitrary partition of block A into block B and some subset of block B into block A. We will perform such moves, but only one vertex at a time, and only if a certain condition is met.

Definition 6: A move of an m-vertex from A to B (B to A) is a proper move only if it covers an (m-1)-vertex which was not previously covered by any member of B (A).

Lemma 5: Starting with an arbitrary initial partition $A|B$ of the m-vertices, some series of proper moves will lead to a member of $P^{-}(m,n)$ if $P^{-}(m,n) \neq \emptyset$.

Lemma 6: Given any member of $P^{+}(m,n)$, if we make any proper move, the result will also be a member of $P^{+}(m,n)$.

Lemma 7: If $P^{+}(m,n) \neq \emptyset$ and $P^{-}(m,n) \neq \emptyset$ then $P^{+}(m,n) \cup P^{-}(m,n) \neq \emptyset$.

Theorem 5: For $m \leq \left\lfloor \frac{n}{2} \right\rfloor$, $R(m) > n$ iff the m-vertices of the n-cube can be partitioned to meet Conditions 1 and 2.

We now present what is known about Ramsey numbers of the type $R(m) = N(m+1,m+1;m)$. It is easy to show that $R(1) = 3$ and relatively easy to show that $R(2) = 6$ [9]. These are the only known Ramsey numbers of this form. Bounds for higher m do exist, however. For $m = 3$ it has been shown [10,11] that $13 \leq R(3) \leq 17$. Bounds for $m > 4$ are rather loose [9], but may still be useful in some instances.

From the preceding paragraph and our previous theorems, several conclusions can be drawn concerning the construction of totally self-checking checkers. For example, the only 1-out-of-n and 2-out-of-n two-level totally self-checking checkers that can be constructed are the trivial 1-out-of-2, the 2-out-of-4, and the 2-out-of-5 checkers. From the bounds on R(3) we can conclude that 3-out-of-n totally self-checking checkers can definitely be constructed in two levels for $6 \leq n \leq 12$, and definitely not for $n \geq 17$. Furthermore, we can reach similar conclusions for (n-1)-out-of-n, (n-2)-out-of-n and (n-3)-out-of-n checkers. From a consideration of coverings we see that it is easy to partition the m-out-of-n codewords for $m = \frac{n}{2}$ so that we meet Conditions 1 and 2, and a relatively large number of such partitions exists. As m becomes smaller (or larger) for a fixed n, fewer and fewer satisfactory partitions exist and it becomes increasingly difficult to find one. Finally, when $R(m) \leq n$ it becomes impossible to find a satisfactory partition.

An m-out-of-n code has an optimal or near optimal number of code words when $m \simeq \left\lfloor \frac{n}{2} \right\rfloor$ [1]. For this reason, m-out-of-n codes with $m \simeq \left\lfloor \frac{n}{2} \right\rfloor$ are of the greatest practical interest. For these particular codes, the designer of totally self-checking checkers is given a great deal of latitude in choosing a design which fits his particular constraints. Any of a large number of partitions of m-vertices into F and G may be used, and given the partition, any inverter-free circuit may be used to realize f and g. Thus, a designer may find that one partition and a particular circuit give him a design with minimal gates, minimal lines, or whatever property he may be interested in. Therefore, although the problem of finding a satisfactory partition of the code vertices into F and G is in general difficult, it is relatively easy for codes of practical interest.

4. SUMMARY AND CONCLUSIONS

Sufficient conditions on the mapping of m-out-of-n code words such that any inverter-free realization of the mapping is a totally self-checking checker are given. Previously given checker designs fall into the class of checkers which meet these conditions.

The subclass of two-level checkers is then considered, and the sufficient conditions are proved to be necessary for two-level realizations. From this we can deduce that our sufficient conditions are as loose as they can be. That is, if a mapping of m-out-of-n words does not meet the sufficient conditions, at least one realization of that mapping will not be a totally self-checking checker.

The sufficient conditions cannot always be met by a particular m and n, and a relationship between m and n using Ramsey numbers is given from which it can be determined whether the conditions can or cannot be met.

The difficulty of showing the presence of the sufficient conditions for an arbitrary m and n and the difficulty of finding minimal two-level designs for arbitrary m and n point out the level of complexity inherent in the structure of m-out-of-n codes. However, this complexity is not prohibitive if we desire m-out-of-n checkers with $m \simeq \frac{n}{2}$, or two-level checkers which are only near-minimal.

REFERENCES

1. Anderson, D.A., "Design of Self-Checking Digital Networks Using Coding Techniques," Coordinated Science Laboratory Report R-527, University of Illinois, September 1971.

2. Anderson, D.A. and G. Metze, "Design of Totally Self-Checking Check Circuits for m-out-of-n Codes," IEEE Trans. Comp., Vol. C-22, p.263-269, March 1973.

3. Carter, W.C. and P.R. Schneider, "Design of Dynamically Checked Computers," IFIP 68, Vol. 2, Edinburgh, Scotland, pp.878-883, August 1968.

4. Chang, H.Y., E. Manning, and G. Metze, Fault Diagnosis of Digital Systems, Wiley-Interscience, New York, 1970.

5. Reddy, S.M., "A Note on Self-Checking Checkers," IEEE Trans. Comp., Vol. C-23, pp. 1100-1102, October 1974.

6. Ho, D., "The Study of a Totally Self-Checking Adder," Coordinated Science Laboratory Report R-582, University of Illinois, August 1972.

7. Betancourt, R., "Derivation of Minimum Test Sets for Unate Logical Circuits," IEEE Trans. Comp., Vol. C-20, pp.1264-1269, November 1971.

8. Ramsey, F.P., "On a Problem of Formal Logic," Proc. London Math.Soc., 2nd Ser., Vol. 30, pp. 264-286, 1930.

9. Liu, C.L., Topics in Combinatorial Mathematics, notes on lectures given at the 1972 MAA Summer Seminar, WilliamsCollege, Williamstown, Mass., Math. Assoc. of America, 1972.

10. Isbell, J.R., "$N(4,4;3) \geq 13$," Journal of Comb. Theory., Vol. 6, p.210. 1969.

11. Kalbfleisch, J.G., "On the Ramsey Number $N(4,4;3)$," Recent Progress in Combinatorics, W.J. Tutte, Ed., Academic Press, New York, pp.273-282, 1969.

12. Bouricius, W.G., W.C. Carter, K.A. Duke, J.P. Roth, and P.R. Schneider, "Interactive Design of Self-Testing Circuitry," Purdue Centennial Year Symp. on Information Processing, pp. 73-80, April 1969.

13. Schönheim, J., "On Coverings," Pacific Journal of Math., Vol. 14, pp. 1405-1411, 1964.

14. Kalbfleisch, J.G. and R.G. Stanton, "Maximal and Minimal Coverings of (k-1)-tuples by k-tuples," Pacific Journal of Math., Vol. 26, No. 1, pp. 131-140, 1968.

15. Smith, J.E. and G. Metze, "General Design Rules for the Construction of m-out-of-n Totally Self-Checking Checkers," Coordinated Science Laboratory Report R-693 , University of Illinois.

A PSEUDOBOOLEAN MULTIVALUED SWITCHING ALGEBRA

JEAN DUSSAULT AND MOSHE KRIEGER
University of Illinois, Urbana, Illinois and
University of Ottawa, Ottawa, Canada

ABSTRACT

This paper introduces a N-valued ($N=2^i$) pseudoboolean algebra obtained by extending the set of operators of N-valued boolean algebras. The additional operators, needed for functional completeness, are unary operators and were selected such that the basic structure and simplicity of boolean algebras are retained. The characteristics of this switching algebra are investigated in detail.

INTRODUCTION

The current theory of multivalued logic is mainly based upon Post algebras [1] and their extensions [2-4]. Some relationships between Post and boolean algebras have been investigated in [5]. More recently an algebra has been proposed with boolean properties which uses the cycling operation [6].

The major drawback of Post's algebra is its lack of simplicity or dissimilarity with the binary boolean algebra. Higher order boolean algebras, which are the simplest extension of conventional switching algebra, are functionally incomplete. But, by restricting the number of truth values to powers of two and making use of a particular unary operator, an algebra which satisfies all the theorems of boolean algebra is obtained.

I - BACKGROUND AND DEFINITIONS

In this section we define and examine the basic properties of the pseudoboolean algebra. To make the section self-contained boolean algebras will be introduced using the concept of lattices. A lattice is a partially ordered set in which every pair of elements a and b has a least upperbound, denoted $a+b$, and a greatest lower bound, $a \cdot b$. All lattices obey the following operational laws [6]:

1) Idempotence: $a+b = a$, $a \cdot a = a$
2) Commutativity: $a \cdot b = b \cdot a$, $a+b = b+a$
3) Associative laws: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$, $(a+b)+c = a+(b+c)$
4) Absorptive laws: $a+a \cdot b = a$, $a \cdot (a+b) = a$

A boolean algebra is a complemented distributive lattice. A lattice is distributive if and only if it satisfies the following:

5) Distributive laws $a+(b \cdot c) = (a+b) \cdot (a+c)$; $a \cdot (b+c) = a \cdot b + a \cdot c$

In a lattice with "0" the least element and "1" the largest, by a complement of x, $\overline{x}$ is meant to be such that:

6) $x \cdot \overline{x} = 0$, $x+\overline{x} = 1$

In a distributive lattice complementation is unique. The 2^i subsets of a set of i unordered elements form a boolean algebra, since every subset has unique complement. Lattice structures and their associated truth tables can be found in [7].

An expression $F(X_1,\ldots,X_n)$ is a boolean function in n variables if it can be constructed using the following statements: 1) Any constant or any variable is a boolean function. 2) Given that $g(X_1,\ldots,X_n)$ and $h(X_1,\ldots,X_n)$ are boolean functions then any combination of g and h under a finite number of applications of the $+$, $\cdot$, $\overline{\ }$ operations is a boolean function. The expansion theorem for such functions follows:

Theorem 1.1:

Given a N-valued boolean function of n-variables $F(X_1,\ldots,X_n)$, it can be expanded as $F(X_1,\ldots,X_i,\ldots,X_n) = F_i(0)\cdot\overline{X}_i + F_i(1)\cdot X_i$ where $F_i(\theta) = F(X_1,\ldots,\theta,\ldots,X_n)$ [9].

Each of the F_i's can be expanded in a similar way. The expansion will eventually give an expansion of the form described in the next theorem:

Theorem 1.2:

Every boolean function of n variables $X_1,X_2,\ldots X_n$ over a N-valued boolean algebra can be expressed in a disjunctive canonical expansion:

$$F(X_1,X_2\ldots X_n) = \sum_{i=0}^{2^n-1} f(i)\cdot m_i \text{ with } m_k=X_1^{\alpha_1}\, X_2^{\alpha_2} \ldots X_n^{\alpha_n},\ X_i^{\alpha_i} = \begin{cases} \overline{X}_i & \text{if } \alpha_i=0 \\ X_i & \text{if } \alpha_i=1 \end{cases}$$

$$\text{and } f(k)=f(\alpha_1,\alpha_2,\ldots\alpha_n) \text{ with } k= \sum_{i=1}^{n} \alpha_i 2^{n-i}$$

The proof of the above theorem can be found in [9]. Its dual expression is $F(X_1,X_2\ldots X_n) = \prod_{i=0}^{2^n-1} (f(i)+\overline{m}_i)$. This theorem states that a n-variable, N-valued boolean function, is completely defined given the values at 2^n points, corresponding to the different combinations of the identity elements or corner points. Also given a N-valued boolean algebra $(N=2^i)$ there are exactly N^{2^n} different boolean functions. It should be observed that a n-variable truth table has N^n rows and in general, the definition of all the functions requires the consideration of N^{N^n} truth tables. These truth tables will be called combinatorial. In general, very few functions are boolean and many more are combinatorial. In fact, this implies that only the 2-valued boolean algebra is complete.

In conclusion, to realize N-valued boolean functions of n variables the $+$, $\cdot$, and $-$ are sufficient. The realization of all combinatorial functions requires the introduction of unary operators and this is basically what is done next.

The algebra now defined is characterized by boolean properties and a very simple set theoretical manipulations of the unary operators. Some basic definitions follow:

Definition 1.1:

The set of N truth values will be denoted by $S=\{0,a_1,a_2\ldots a_j,1\}$ where $j=N-2$. Moreover, N will be defined only for i such that $N=2^i$. Truth values will be denoted by ordinary alphabetic characters and set of these by capital letters.

Definition 1.2:

The set of binary operators $O=\{+,\cdot\}$. The elements of O are such that they satisfy the following laws: idempotence, associativity, commutativity, absorption and distributivity. At this point also defined is the inverse $\overline{X}$ such that $\overline{X}$ satisfies the null and universal elements properties.

Now by a well known theorem [9], it is demonstrated that such an algebra has a boolean lattice structure.

Thus the binary operators + and · are the binary lattice operator least upper bound and greatest lower bound and the complementation operation is the same as the lattice complement with the 0 and 1 being the null and identity elements. Thus truth tables will be those referred to before. For functional completeness the following unary operator will be defined:

1.3) the unary operator $X^a=\{ {1 \text{ if } X=a \atop 0 \text{ otherwise}}$.

1.4) Its complement $\overline{X^a} =\{ {0 \text{ if } X=a \atop 1 \text{ otherwise}}$. It is not $X^{\overline{a}}$ but X^D where $D=S-\{a\}$.

These 2 operators, restrained in value to 0 or 1, satisfy all the composition rules listed earlier. The operator X^a is similar to the ${}^aX^a$ of [3] but it is not concerned with range of superscripts and thus is much more general and simpler to use. Indeed superscript manipulations now become trivial set operations. Some properties of the unary operators are now outlined.

i) Properties over addition "+" (lub):

The "sum" operation performed on two unary operators of the same variable is defined to be the variable with superscripts being the set union of each of the superscripts, i.e., $X^i+X^j=X^{i\cup j}$ and will be denoted by $X^{i,j}$ where $X^{i,j}$ is one if and only if X is equal to i or j and 0 otherwise. More generally:

$$\sum_{\forall i} X^{A_i} = X^C \text{ where } C=\bigcup_{\forall i} A_i$$

No similar procedure can be applied if the variables are mixed.

P1) if $A\subset B$ then $X^A+X^B=X^B$ since $A\cup B=B$

P2) from the complement definition $X^A+X^{\overline{A}}=X^S$ since $A\cup S\cap\overline{A}=A\cup\overline{A}\cap S=S$. The literal X^S is the constant "1". It can therefore be ignored in a product and causes all other terms to be redundant in a sum.

ii) Properties over the product "·"(glb):

The product "·" operation, performed on unary operators of the same variable, is defined to be the same variable with superscripts being the set intersection of each of the superscripted subsets. In general:

$$\prod_{\forall i} X^{A_i}=X^D \text{ with } D=\bigcap_{\forall i} A_i$$

If the superscripted subsets are disjoint, then the intersection results in $\emptyset$ where $X^\emptyset$ is the constant "0". Indeed, since the logical product of two identical variables with different truth conditions is always "0", it follows that $X^{\emptyset}$ is "0". The following is a list of some other properties:

P3) $X^A \cdot X^S = X^A$ since $A \cap S = A$ and in general if $A \subset B$ then $X^A \cdot X^B = X^A$

P4) $X^A \cdot X^{\overline{A}} = X^{\emptyset} = 0$ for all A.

All theorems of boolean algebra hold for the above algebra. This is because set manipulations follow the same rules of composition as a boolean algebra (provided the proper translation is done between operators and identity elements). As an example it is proved in [10] that De Morgan's laws hold for this type of algebra.

From the above properties we obtain also that

$$\overline{c \cdot X^i} \begin{cases} \overline{c} & \text{for } X=i \text{ since } \overline{c \cdot X^i} = \overline{c} + \overline{X^i} = \overline{c} + 0 = \overline{c} \\ 1 & \text{otherwise since } c+1=1 \end{cases}$$

Also the complement of a product term is of the form:
$F = (\overline{c}_k + \overline{X_1^{n_{k1}}} + \overline{X_2^{n_{k2}}} + \ldots + \overline{X_n^{n_{kn}}})$ so that by duality we have the product of sum realization.

II N-VALUED SWITCHING FUNCTIONS

Now that basic definitions and elementary properties have been outlined, it is possible to introduce the format of the switching function that will be used. Switching functions will be given in the form of truth tables as shown below consisting of N^n entries, n being the number of variables.

	X_1	X_2	...	X_n	$F(X_1, X_2 \ldots X_n)$
ρ_o	0	0	...	0	α_1
	.	.		.	.
	.	.		.	.
	.	.		.	.
ρ_i	r_{i1}	r_{i2}	...	r_{in}	α_i
	.	.		.	.
	.	.		.	.
	.	.		.	.
ρ_{N^n}	1	1	...	1	α_{N^n}

Table II-1 N-valued truth table

The left side lists input combinations, with a truth value associated with each of the n-tuples. In Table II-1, all r_{ij} and α_j are elements of S, the set of truth values. Since $\alpha_i \cdot 0 = 0$ for all i, then (for row i):

$$F(X_1, X_2 \ldots X_n) = \alpha_i \cdot X_1^{r_{i1}} \cdot X_2^{r_{i2}} \cdot \ldots \cdot X_n^{r_{in}}$$

A product term has the form $i \cdot \emptyset_i$, where i is a constant and $\emptyset_i$ a product sequence of unary operators, in which each variable appears at most once. This product term is much simpler than the one used in [3] since there is no such thing as repetition of variables and the concept of separability is not involved. From this, it follows that a function with exactly one non-zero entry can be defined by a single product term. More generally we have that the canonical form of a switching function of which $m \leq N^n$ entries are non-zero is given by the summation of m product terms. This means that a general switching function can be expressed as:

$$F(X_1,X_2,\ldots X_n) = \sum_{j=1}^{m} \alpha_j \cdot \prod_{i=1}^{n} X_i^{r_{ji}} \tag{2.1}$$

and in complemented form:

$$\overline{F(X_1,X_2,\ldots X_n)} = \prod_{j=1}^{m} \overline{(\alpha_j} + \overline{\sum_{i=1}^{n} X_i^{r_{ji}})} \tag{2.2}$$

A product of sum representation is possible and done in a standard way; i.e., with all the truth values complemented , the sum product representation of $\overline{F(X_1,X_2\ldots X_n)}$ is first obtained. De Morgan's law is then applied and the result gives $F(X_1,X_2\ldots X_n)$.

In order that a N-valued switching function be realizable, it must first be rewritten in an appropriate format. This is what has just been done. It has been constructed by looking at the truth table and found to yield a result similar to the Shannon's expansion theorem for the pseudo binary algebra described in [3]. This result can now be put in the following format:

First decomposition theorem: Given an n-variable switching function $F(X_1,X_2\ldots X_n)$, it can be represented by either one of the following:

$$F(X_1,X_2\ldots X_n) = \sum_{j=0}^{N-1} j \cdot F_j \tag{2.3}$$

$$F(X_1,X_2\ldots X_n) = \prod_{j=0}^{N-1} j+G_j \tag{2.4}$$

where F_j= is a sum of product terms for which the function $F(X_1,X_2\ldots X_n)$is j and G_j= is the complement of F_j.

The proof is a direct consequence of (2.1) and (2.2) since we have grouped the m products (2.1) or sum (2.2) terms into N subfunctions (1 for each truth value). As an example a four and 8-valued functions can be expressed as:

$$F_4(X_1\ldots X_n)=0\cdot F_0+a\cdot F_a+b\cdot F_b+c\cdot F_c$$

$$F_8(X_1,\ldots X_n)=0\cdot F_0+a\cdot F_a+b\cdot F_b+c\cdot F_c+d\cdot F_d+e\cdot F_e+f\cdot F_f+1\cdot F_1$$

Each F_i is a two-valued subfunction of F and each entry of F_i is equal to one, if and only if the corresponding entry of F is equal to i. The above expansion is much simpler than the one of Post's algebra. A canonical form in Post algebra is long, complicated and in unreduced form (even in reduced) messy. The algebra introduced in [3] is simpler in this respect. In the algebra introduced here an interesting property of boolean lattices is used to reduce these canonical expressions. That is that every 2^i-valued combinatorial table can be reduced to the combination of i two-valued subfunctions each preceeded by a constant. This states that the canonical expression of 2^i-valued function will contain at most i terms versus as many N. Thus we obtain:

Second decomposition theorem: Given a n-variable, N-valued ($N=2^i$) switching function $F(X_1,X_2...X_n)$ it can be represented by either one of the following.

$$F(X_1,X_2,...X_n)= \sum_{j=1}^{\log_2 N} \alpha_j \cdot F\alpha_j \tag{2.5}$$

or

$$F(X_1,X_2,...X_n)= \prod_{j=1}^{\log_2 N} \alpha_j + G\alpha_j \tag{2.6}$$

where α_j is one of the lower i truth values

$F\alpha_j$ is a sum of product terms for which the function $F(X_1,X_2...X_n)$ is α_j or $\geq \alpha_j$ (lattice ordering)

$G\alpha_j$ is a product of sum terms for which the function $F(X_1,X_2...X_n)$ is not α_j or $\not\geq \alpha_j$

The proof is obtained by inspecting the boolean lattice structure. In a lattice with 2^i elements, the null element is a common lower bound of the first i elements which are in turn the lower bound (some of them the greatest) of all upper situated atoms. Thus each $j \geq i$ element can be defined as a unique least upper bound consisting only of the i lower atoms. For example, in the 4-valued case:

$a+b=1$, thus $F=a\cdot F_a+b\cdot F_b+1\cdot F_1=a\cdot F_a+b\cdot F_b+(a+b)\cdot F_1$

$=a\cdot(F_a+F_1)+b\cdot(F_b+F_1)$ (consists of 2 terms)

For 8-values:

$d=a+b$, $e=a+c$, $f=b+c$, $1=d+e+f=a+b+c$

the function $F=a\cdot F_a+b\cdot F_b+c\cdot F_c+d\cdot F_d+e\cdot F_e+f\cdot F_f+1\cdot F_1$

$=a(F_a+F_d+F_e+F_1)+b(F_b+F_d+F_f+F_1)+c(F_c+F_e+F_f+F_1)$

$=a\cdot H_a+b\cdot H_b+c\cdot H_c$ (3terms)

This is the "bottom up" approach. It is possible to start from the top of the lattice and obtain similar results with a "top-down" approach. This is due to the perfect symmetry of the lattice (De Morgan's laws hold). It gives a product of sums representation, using the i elements immediately below the universal "1".

In this section a pseudo-boolean algebra was developed by including unary operators to produce a complete set along with the boolean operators. These operators are similar to the Post's system, but they were selected such that they preserve the general boolean characteristics. Using these operators a decomposition theorem was developed similar to the pseudo-binary algebra introduced in [3-4]. It was proved that using the boolean lattice characteristics of our algebra, a simpler decomposition theorem could be obtained. This decomposition theorem uses $i=\log_2 N$ subfunctions instead of N subfunctions and it can be considered as a generalization of the boolean function decomposition theorem. The next section will present realization and minimization techniques for the system.

III REALIZATION AND MINIMIZATION

This section will first review the decomposition theorems, this time with respect to the realization of subtables and networks. Then cost functions will be discussed for the different realization structures and finally a minimization procedure will be outlined.

Section II indicated two decompositions of an N-valued switching function. The following is concerned with the decompositions of the form of equations 2.3 and 2.5. Corresponding to these decompositions, the following realizations both correspond to multiple output networks:

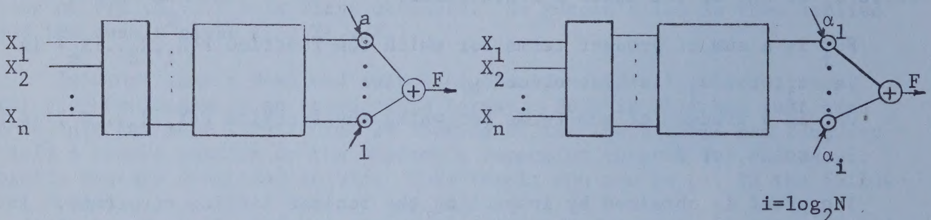


Fig. 3.1 and 3.2: Two realizations corresponding to (2.3) and (2.5)

From a function in a N-valued field, there are as many as N subfunctions and they correspond to the first decomposition. These subfunctions are $F_a, F_b, \ldots F_1$. They can be grouped in $\log_2 N$ subfunctions, which are sufficient to represent the function, since it has been shown that the previously defined lattice has a number of truth values that can be reduced to the combination of $i=\log_2 N$ truth values. This is the second decomposition and gives i subfunctions denoted by $F\alpha_1$, $F\alpha_2 \ldots F\alpha_i$. Note that each of the subfunctions F_j and $F\alpha_j$ are basically two valued functions and this can be defined by 2-valued subtables. There follows a formal way (algorithms) to obtain these subtables.

i) The construction of F_k subtables: For an input combination ρ_j the corresponding entry in subtable F_k denoted α_{k_j} will be: a) "1" if and only if $F(\rho_j)$ is k; b) "d" if it is a don't care condition; c) "0" otherwise. The tables thus produced are disjoint except maybe in "don't care" coordinates.

ii) The construction of $F\alpha_k$ is as follows: For an input combination ρ_j the corresponding entry is subtable $F\alpha_k$ denoted α_{k_j} will be: a) "1" if $F(\rho_j)=\alpha_{k_j}$ or if $F(\rho_j)=\beta_j$ and $\alpha_{k_j} \leq \beta_j$; b) "d" if it is a don't care position; c) "0" otherwise.

That is, β_j has been obtained from the sum (l.u.b.) operation on α_{k_j} with other α_j's. This produces $i=\log_2 N$ non disjoint tables. Supposing there are K non-zero entries in a N-valued function's table and there are L entries which take on values equal to the first i values of the ordered set of truth values, then all other (K-L) entries must produce a "1" in at least two subtables. By duality, subtables can be built using only the topmost i elements of the lattice. Also, from an other point of view, the functions

$F\alpha_k$ can be obtained by performing a union of appropriate subtables F_k and vice-versa, using the intersection operation. The next step is to consider the different network realization of a given function.

Having established the overall network structure, (Fig. 3.1 and 3.2) we now examine some concepts involved in the realization of such structures. The number of product gates in Fig 3.1 will be larger than in the second representation (Fig. 3.2). The first "+ gate" will also be much more complex than the "+ gate" corresponding to the second decomposition. This is due to the fact that the summing gate in (ii) above can be replaced by a regular summer, since there are not two inputs that have the same value nor be the complement of the other. Also, all the gates in Fig. 3.2 could be replaced by a simple weighted summer with at most $i=\log_2 N$ inputs. From the structure of Fig. 3.1 to that of Fig. 3.2, the number of gates has been reduced from N+1 to i+1. This reduction gives a saving of 2^i-i product gates. Not only is it more economical to use the configuration of Fig. 3.2 but also, since there are fewer subfunctions there will be more "1"s in each of the subfunctions and it is probable that there will be more simplification. Therefore only networks with this configuration will be used.

Now, if the system were in regular binary boolean algebra, the overall problem of realization would be solved since the gating introduced would be sufficient to realize all the subfunctions. However, because the system is in multivalued logic, some sort of decoder has to be used. The separation shown in Fig. 3.2 is only conceptual and in the actual physical implementation, it may be impossible to define a boundary between the decoder and the binary logic circuit.

The decoder has multivalued input signals and its outputs are restricted to "0" and "1". Two possible structures are of the form X^a and $X^{a,b\ldots}$, giving the decoder realizations of Figure 3.3.

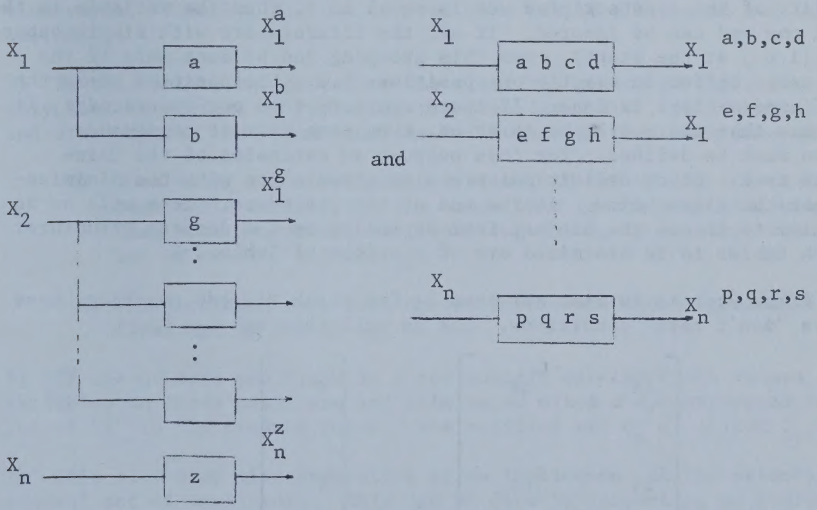


Fig. 3.3: Possible decoder constructions

In the first case, only variables with one superscript are authorized and only those that are actually needed are constructed. In the second case, some logic is already done within the decoder and this emphasizes the statement concerning the existence of a separate decoder.

Now that a network structure has been established, the cost of physically implementing it will be briefly discussed.

In multiple valued logic, the situation which prevails is somewhat different from binary logic because the overall cost function is strongly dependent on the use of unary operators. They will be assumed to be more expensive to construct than conventional gates. The gates of the "summer" are already minimal and represent only a fixed cost which is directly proportional to $\log_2 N$. Thus as was expected, the minimization problem for multiple valued logic systems is related to that of a 2-valued system, but it has differences. We have assumed that the summer is standardized and the binary circuit inexpensive in the sense that a minimal decoder construction should give a good enough binary circuit.

In general the type of construction that will give minimal realization is not evident, since the difference between the relative cost of implementing X^a and $X^{a,b\cdots}$ is not evident and is strongly dependent on the actual electronic implementation (see [10]). In conventional logic, the study of prime implicants gives directly a minimal cost representation. Fortunately, the extension of this, in our case, is also valid, while it is not in some other multivalued algebras. Trivial operations on the set of prime implicants produce minimal solutions according to the cost criterion. Since prime implicants are a major concern, a technique to obtain them will be derived next.

A map method is described in [10] but is a very limited approach to minimization so we only give the algebraic manipulations behind the process. The main idea, is to eliminate literals whether they are of the form X^i or X^A. The grouping of literals is possible, because $X^A + X^B = X^{A \cup B}$. If the cardinality of the superscripted set is equal to N, then the variable is the constant one and can be ignored. If all the literals are with simple superscript, (i.e., at the start), then this grouping can be done only if the product terms differ in exactly one position, i.e., the position where the union of superscripts is done. If there are more than one superscript, it is probable that there will be later on, then some form of consensus operation must be defined. For this purpose an extension of the Quine method is used. Other definitions are also given along with the minimization algorithm given below. At the end of the procedure, there will be an opportunity to choose the minimal form depending on the decoder structure. The truth tables to be minimized are of the form of Table 2-2.

1) The first step is to list all rows ρ_j for which $F(\rho_j) \neq 0$, marking those which are "don't care" conditions. Let us call this set C_o.

$$C_o \equiv \begin{bmatrix} r_{1s} & \cdots\cdots & r_{ns} & \\ \cdot & & \cdot & \\ \cdot & & \cdot & \\ \cdot & & \cdot & \\ r_{1t} & \cdots\cdots & r_{nt} & \\ & & \cdot & \\ r_{1v} & \cdots\cdots & r_{nv} & (d) \\ \cdot & & \cdot & \\ \cdot & & \cdot & \\ \cdot & & \cdot & \\ r_{1w} & \cdots\cdots & r_{nw} & (d) \end{bmatrix} \equiv \begin{bmatrix} \rho_s & \\ \cdot & \\ \cdot & \\ \cdot & \\ \rho_t & \\ & \\ \rho_v & (d) \\ \cdot & \\ \cdot & \\ \cdot & \\ \rho_w & (d) \end{bmatrix}$$

2) Next perform the consensus operation between every pair of rows ρ_t and ρ_e

Definition: The consensus in the i^{th} coordinate, between two rows $\rho_t=r_{1t}r_{2t}\cdots r_{nt}$ and $\rho_e=r_{1e}r_{2e}\cdots r_{ne}$, denoted $\rho_{1x}=r_{1x}r\ \ldots r_{nx}$ is defined by:

1) $r_{ix}=r_{it}\cup r_{ie}$ for one i between 1 and n

2) $r_{jx}=r_{jt}\cap r_{je}$ for all $j\neq i$

If the result of the intersection is $\emptyset$ then the consensus in the i^{th} position does not exist; otherwise the consensus is ρ_x.

So far as the minimization procedure is concerned the consensus in all n positions must be performed. Also note that "d" is conserved under the union operation but is removed under the intersection operation, if only one of the two elements has a "d" attached to it. Parentheses are used to save the results from the union operation. The set of all consensus will be called the complex C_1.

3) Both set C_1 and C_o are merged and the result C_2 is: $C_2=C_o\cup C_1$.

4) The purpose of this step is to eliminate redundancy. For this reason it is important to define what is meant by coverage.

Definition: A row $\rho_\theta=r_{1\theta}r_{2\theta}\cdots r_{n\theta}$ is said to be covered by a row $\rho_\eta=r_{1\eta}r_{2\eta}\cdots r_{n\eta}$ if and only if $r_{i\theta}\cap r_{i\eta}=r_{i\theta}$ for all i.

In general, rows from which consensus has been obtained are covered so they can be marked as the tests proceed so that fewer tests need be performed. Note that a row can always cover a row with don't care conditions but not vice-versa. Finally, all the covered rows can be eliminated and the result called C_3.

5) To this point, the procedure described has generated implicants. Now they must be refined by repeating the procedure, i.e., set C_3 equal to C_o and start again. But this time the following must be taken into account:

i) When doing the consensus operation, if one of the intersection results is smaller (in the sense that it contains less elements than both operands), the resulting row ρ_x is usually redundant and can be readily eliminated.

ii) The procedure ends when no new implicants can be generated. The final set is called C_x.

6) If one or more positions in a row contain all the truth values, the variables in these positions are eliminated since they correspond to a factor of "1" in the product terms. The modified set C_x is called C_y.

7) This last step (for generating prime implicants) is the selection of a minimal set of implicants. This can be done by inspecting an implicant table (constructed in a standard way) and selecting the minimal number of vertices that covers the functions. From this minimal set, delete dominated columns which, from an algebraic point of view, is the same as performing $C_{yj}\cap C_{yk}$ on all columns. Then if the result is not $\emptyset$ in at least one position, it is possible to eliminate the result of the operation from one row

of the pair C_{y_j} and C_{y_k} (but not both). The row, from which it is removed, contains the dominated column and overlapping of the implicant is avoided.

The algorithm has now generated all the prime implicants. Depending on the cost function or on the decoder construction, it may be necessary to perform an additional step. If the decoder allows more than one superscript per variable, then the set obtained is the minimal set and can be complemented directly. However, if the decoder is to be made up of simpler blocks blocks, i.e., of the form X^a, take the union of all values column by column and the resulting set will give the minimum set of X_j^i that must be produced.

CONCLUSION

The main purpose of this paper is the development of an N-valued switching algebra, with characteristics similar to a boolean algebra in structure as well as in simplicity. To construct such a system, it was necessary to restrict the number of truth values to some power of two and to introduce auxiliary operators. These auxiliary operators are unary operators, similar to those defined in other multivalued algebras, such as the system of Post. These were selected such that the boolean properties were preserved and manipulations were simple.

It is shown that, using the binary operators of N-valued boolean algebras and the chosen unary operator, switching functions are easy to express and expand. Indeed, two expansion theorems are presented, the first being similar to the theorem encountered in the literature of general N-valued logic and the second being a direct consequence of the boolean properties of the proposed algebra. This second theorem reduces the number of subfunctions to $i=\log_2 N$, while other systems can have as many as N subfunctions. Such an expansion theorem was found to provide simple realization and minimization. Indeed, a relatively simple algorithmic minimization technique was presented.

REFERENCES

1. E. L. Post, "Introduction to a General Theory of Elementary Propositions", American Journal of Mathematics, vol. 43, pp. 163-185, 1921.
2. Z. G. Vranesic, E. S. Lee, K. C. Smith, "A Many-Valued Algebra for Switching Systems", IEEE Trans. on Computers, vol. C-19, no. 10, pp. 964-971, October 1970.
3. D. D. Givone and R. W. Snelsire, Final Report on the Design of Multiple Valued Logic Systems, State University of New York at Buffalo, Buffalo, 1968.
4. C. M. Allen and D. D. Givone, "A Minimization Technique for Multiple Valued Logic Systems", IEEE Trans. on Computers, Vol. C-17, no. 2, pp. 182-184, February 1972.
5. A. S. Wojcik and G. Metze, "Some Relationships Between Post and Boolean Algebras", in Proc. of the 1971 Symposium on Theory and Applications of Multivalued Logic Design, May 27-28, 1971, pp. 173-182.
6. S. C. Lee and Y. Keren-Zvi, "A Generalized Boolean Algebra and its Application to Logic Design," in Proc. of the 1975 Inter. Symp. on Multiple-valued Logic, May 13-16 1975, pp. 88-98.
7. R. Dubisch, Lattice to Logic, Blaisdell, New York, 1964.
8. S. A. Kiss, Transformations on Lattices and Structures of Logic, Stevens & Co., New York, 1947.
9. R. Prather, Switching Theory, a Mathematical Approach, Allyn Bacon, 1967.
10. J. Dussault, "A Pseudoboolean Multivalued Switching Theory", M. Sc. A. Thesis, University of Ottawa, Ottawa, July 1974.

MULTI-VALUED ASYNCHRONOUS MODULES

ANTHONY S. WOJCIK
Department of Computer Science
Illinois Institute of Technology
Chicago, Illinois 60616

ABSTRACT

The use of multi-valued signals in the design of asynchronous speed-independent modules is considered. Basic data/control signal detectors and a basic control module are developed. A general control structure that allows the interconnection of complex combinational modules is described. Extensions to sequential modules are discussed.

INTRODUCTION

A topic of considerable interest in the field of switching theory has been the analysis and synthesis of asynchronous networks. In particular, the design of "speed-independent" asynchronous circuits has been studied by a number of researchers including [1-5]. A major problem in speed-independent design has always been the inclusion of control signals that are required to allow the movement of data between speed-independent modules and to control the data-processing function of the modules. Several control schemes have been proposed, with one of the most interesting described by Armstrong, et al [4,6] which uses encoded data and control signals.

The purpose of this paper is to describe a technique for the utilization of multi-valued signals in the design of speed-independent asynchronous modules. A few attempts at using non-binary signals have been reported [2,3,7,8], but no results having general applicability appear to have been reported.

BASIC PRINCIPLES

The multi-valued approach reported in this paper is based on the following interpretation of three-valued Post algebra [9]. If the three logic values of the Post algebra are denoted 0,1 and 2, with the ordering $0 < 1 < 2$, the 0 and 2 values are treated as the standard binary information values while the intermediate 1 value is used as a control level. The Post algebra operations of MAX, +, and MIN, $\cdot$, are the same as the two-valued Boolean operations of OR and AND with respect to the 0 and 2 values. Two-valued Boolean complementation is preserved by using the three-valued strong-negation operation $\overline{X} = 2 - X$ where $X \in \{0,1,2\}$. Table 1 summarizes these three basic operations.

Multi-valued asynchronous operation is restricted to signal behavior in which any change from 0 to 2 (2 to 0) is forced to pass through the intermediate control level. This characterization of asynchronous operation allows for the design of hazard-free combinational networks and the use of technologically feasible logic elements [10,11].

If we assume that a network is initially set to the control level, the only signal changes that can occur when the $+$, $\cdot$, and $\overline{}$ operators are used are those from 1 to 0 (0 to 1) and 1 to 2 (2 to 1). Hence, if we consider a term of a logic expression that is formed by the MIN of a set of literals $X_1, \ldots, X_n$, where the X_i may be in true or complemented form, setting all the literals initially to 1 forces the term to be 1. As literals change from the control level to the binary information levels of 0 or 2, the term will then change value either to 0 or to 2. If an expression is composed of the MAX of a collection of terms, it can be seen that the expression will initially have the value 1 and will change value either to 0 or to 2. It follows, then, that based on the given interpretation of the logic values 0, 1 and 2 and the characteristics of the $+$, $\cdot$ and $\overline{}$ operators, logic expressions for combinational networks can be generated by using standard Boolean design considerations. The imposition of the control level does not require a non-binary design process.

Based on these initial considerations, let us turn to the description of the structure of the asynchronous modules.

MODULE STRUCTURE

Figure 1 depicts the basic structure of a module. This general structure shows that the module is provided with output storage, control, and the ability to receive and to transmit control information to other modules. The speed-independent assumption that logic gates operate with finite, but unbounded, delay and that line delay is zero are assumed. It is further assumed that inputs to a module are provided in both true and complemented form. This is easily accomplished since if we consider that a module input is the output of some other module, the module output will be stored in a three-valued version of the JK flip-flop. This device, shown in Figure 2, has the important characteristic that when it is in the 1 state, both of its outputs, y and $\overline{y}$, are at the 1 level, corresponding to the definition of three-valued strong negation.

The two basic problems that arise in a network composed of such modules are determining when new data is present on the inputs to a module and determining when storage may be updated with new information. We will not be concerned with the initial data source for the network.

A standard approach is used to distinguish between new data inputs, namely, a special "spacer" is used. Since the 0 and 2 levels are used to represent data, the 1 level is used as the spacer control signal. Hence, if we consider the n input lines to a module as an input word, the word consisting of all 1s is the spacer word while data words must consist only of 0s and 2s. Hence, the control portion of a module must recognize the spacer word. To accomplish this, some special logic gates will be incorporated that are three-valued.

Let us define unary operators as follows:

$$f(X) = X^{(ijk)} = \begin{cases} k \text{ if } X = 0 \\ j \text{ if } X = 1 \\ i \text{ if } X = 2 \quad \text{where } i,j,k \in \{0,1,2\} \end{cases}$$

Hence the identity operator is $X = X^{(210)}$ while the strong-negation operator is $\overline{X} = X^{(012)}$. Unary operators are represented in logic diagrams as in Figure 3. The four operators defined in Table 2 will be used as building blocks in the control portion of the modules.

It is essential to recall that the assumed characterization of asynchronous operation requires that any variable, or operator output, that is to change from 0 to 2 (2 to 0) must pass through the 1 level. For example, if X successively takes on the values 0, 1 and 2, the operator $X^{(200)}$ would respond with the output values 0, 0, 1 and 2. The generation of the 1 value is required to allow the control to function correctly.

In order to update storage within a module, control will be provided with information from successor modules concerning their status. Basically, a module can be in any one of three conditions:

1. The module has processed the spacer word and can accept a data word input.
2. The module has processed a data word and is waiting to update its storage.
3. The module has updated its storage and can accept the spacer word.

It can be seen, then, that correct operation of a network of modules requires the presence of a spacer word in at least one module separating modules that are processing data words. Module status information provides the means for internally synchronizing the movement of data/spacer words.

CONTROL PRIMITIVES

Three control primitives are used in the control portion of a module.

1. D primitive. $D(X,Y) = X^{(020)} \cdot Y^{(020)} + 1 \cdot (X^{(020)} + Y^{(020)})$. This operator is described in Figure 4. Its characteristics are that it generates the value 2 only when both inputs are at 1, and it generates the value 0 only when both inputs are at binary information levels.
2. F primitive. $F(X,Y) = X^{(210)} \cdot Y^{(210)} + 1 \cdot (X^{(210)} + Y^{(210)})$. This operator is described in Figure 5. Its major characteristics are that it generates the value 0(2) only when both inputs are at 0(2), otherwise it generates the value 1.

Figure 6 illustrates the use of these primitives. It can be seen that the M input to the module control will be a 2 when all module inputs X_i, $\overline{X}_i$ are at 1 and when the components of the module output function, f, are at 1. As the module inputs take on data values and as the module network processes the data, M will become a 1 - indicating that data is being processed - and then a 0 - indicating that all input data is present and has been processed.

3. CONTROL primitive. This network receives as inputs the M and f lines of Figure 6, a C_s input describing the status of successor modules, and a y input indicating the status of the module storage element. This primitive is described in Table 3 and Figure 7. It controls the updating of the storage element via the JK equations

$$J = 1 \cdot M^{(200)} \cdot f^{(210)} \cdot C_s^{(002)} + y^{(220)} \cdot M^{(012)} \cdot f^{(210)} \cdot C_s^{(210)}$$
$$K = 1 \cdot M^{(200)} \cdot f^{(012)} \cdot C_s^{(002)} + y^{(022)} \cdot M^{(012)} \cdot f^{(012)} \cdot C_s^{(210)}.$$

Figure 7 also depicts the formation of the status signal, C_p, that is sent to the module's predecessors.

The CONTROL primitive's operation is fully depicted in Table 3. The blank entries in this table are input conditions that can never occur. The initial operating point of the CONTROL primitive is at (y M f C_s) = (1212). Succeeding operating points for the primitive occur only according to the asynchronous mode of operation described - particularly, all changes of signal from 0 to 2 (2 to 0) pass through the 1 level. The JK equations are designed to insure hazard-free operation and to reflect the appropriate behavior of the CONTROL under all the possible input conditions - which reflect the stated delay assumptions - that can occur.

SEQUENTIAL MODULES

The ability to use standard Boolean design procedures in forming combinational data processing functions greatly simplifies the design of complex modules. By appropriate use of the D, F and CONTROL primitives, modules can be interconnected in a straightforward manner. Although only a small amount of non-binary design is required, the power of the intermediate control level can be utilized.

Currently, work is being conducted on the incorporation of sequential data processing functions in the modules. The goal of this work is to be able to use standard Boolean sequential design procedures to form ternary sequential functions. This investigation looks quite promising and should result in a process for the design of asynchronous modules that is all inclusive.

CONCLUSIONS

The results that have been reported in this paper clearly demonstrate the feasibility of incorporating multi-valued signals in the design of asynchronous speed-independent modules. Further study is needed to determine if alternate design approaches using non-binary devices might yield a simpler control structure, and to consider other approaches to the data/spacer word technique to differentiate between new and old data words.

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation, Division of Computer Research, under Grant DCR 74 - 01409.

REFERENCES

[1] D.E. Muller, "Asynchronous logics and application to information processing," Proc. Symp. on Application of Swit. Theory in Space Technology, Aiken and Main, Eds., Stanford University Press, Stanford, California, pp. 289-297, 1963.

[2] D. Hammel, "Ideas on asynchronous feedback networks," Proc. 5th Ann. Symp. on Swit. Circuit Theory and Logical Design, Princeton University, pp. 4-11, November 1964.

[3] B. Elspas, et al., "Investigation of propagation-limited computer networks," Final Reports - Phases I,II,III, Stanford Research Institute, April 1964, July 1965, June 1966.

[4] D.B. Armstrong, A.D. Friedman, and P.R. Menon, "Design of asynchronous circuits assuming unbounded gate delays," IEEE Trans. Comp., Vol. C-18, pp. 1110-1120, December 1969.

[5] R.M. Keller, "Towards a theory of universal speed-independent modules," IEEE Trans. Comp., Vol. C-23, pp. 21-33, January 1974.

[6] S.H. Unger, Asynchronous Sequential Switching Circuits, Wiley, New York, Chapter 6, 1969.

[7] A.S. Wojcik, "An application of ternary logic to logic sequencing," Coordinated Science Laboratory Report R-388, University of Illinois, Urbana, July 1968.

[8] H. Mine, et al., "Self-timing asynchronous logical networks using 3-valued NAND circuits and its effect on a binary adder," Report of the Faculty of Eng., Kyoto University, Kyoto-shi, Japan, 1968.

[9] A.S. Wojcik and G. Metze, "An analysis of some relationships between Post and Boolean algebras," J.ACM, Vol. 21, Number 4, pp. 680-696, October 1974.

[10] A.S. Wojcik, "Multi-valued asynchronous circuits," 1973 Int. Symp. on Multiple-Valued Logic, University of Toronto, May 1973.

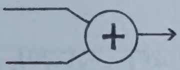
[11] A.S. Wojcik, "Multi-valued asynchronous sequential circuits," 1974 Int. Symp. on Multiple-Valued Logic, West Virginia University, May 1974.

Table 1. Basic logic operations.

+	Y		
X	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

•	Y		
X	0	1	2
0	0	0	0
1	0	1	1
2	0	1	2

X	$\overline{X}$
0	2
1	1
2	0



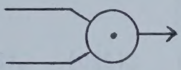


Figure 1. Module structure.

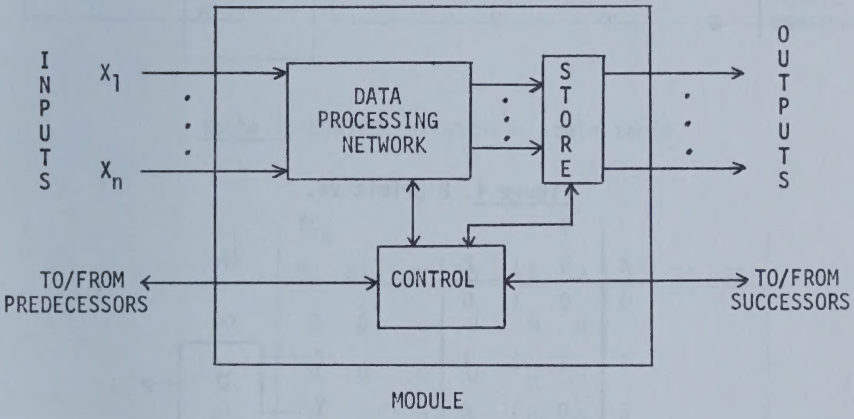
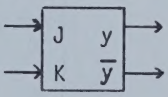

INPUTS
X_1
X_n
DATA PROCESSING NETWORK
STORE
OUTPUTS
CONTROL
TO/FROM PREDECESSORS
TO/FROM SUCCESSORS
MODULE

Figure 2. Three-valued JK flip-flop.

y	JK								
	00	01	02	10	11	12	20	21	22
0	0	0	0	1	1	1	2	2	2
1	1	1	0	1	1	1	2	1	1
2	2	1	0	2	1	0	2	1	0


J
K
y
$\overline{y}$

$$Y = J \cdot \overline{K} + \overline{K} \cdot y + J \cdot \overline{y}$$

Figure 3. Unary operators.

$$X^{(ijk)} : X \rightarrow \boxed{i\ j\ k} \rightarrow$$

$$X^{(210)} : X \longrightarrow$$

Table 2. Building block operators.

X	$X^{(002)}$	$X^{(022)}$	$X^{(220)}$	$X^{(200)}$	$X^{(020)} = X^{(220)} \cdot X^{(022)}$
0	2	2	0	0	0
1	0	2	2	0	2
2	0	0	2	2	0

Figure 4. D primitive.

	Y		
X	0	1	2
0	0	1	0
1	1	2	1
2	0	1	0

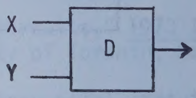


Figure 5. F primitive.

	Y		
X	0	1	2
0	0	1	1
1	1	1	1
2	1	1	2

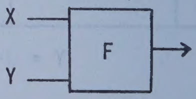


Figure 6. Use of D and F primitives.

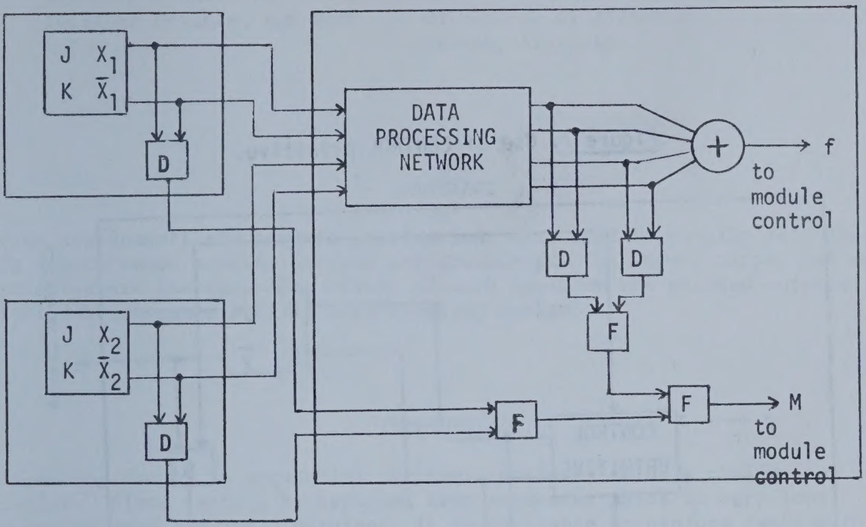
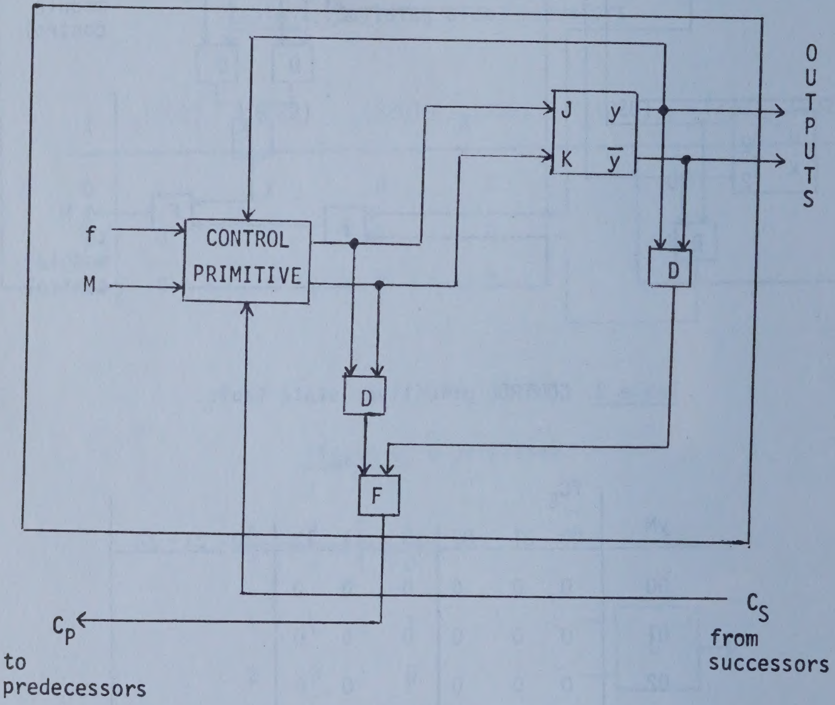

J
K
X_1
$\bar{X}_1$
D
J
K
X_2
$\bar{X}_2$
D
DATA PROCESSING NETWORK
+
f
to module control
D
D
F
F
F
M
to module control

Table 3. CONTROL primitive: state table.

	fC_S								
yM	00	01	02	10	11	12	20	21	22
00	0	0	0	0	0	0			
01	0	0	0	0	0	0			
02	0	0	0	1	0	0			
10	1	1	0	1	1	1	1	1	2
11	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1
20				2	2	2	2	2	2
21				2	2	2	2	2	2
22				1	2	2	2	2	2

Y
(New state of storage)

Figure 7. Use of CONTROL primitive.


OUTPUTS
J
y
K
$\overline{y}$
f
M
CONTROL PRIMITIVE
D
D
F
C_S
from successors
C_P
to predecessors

DESIGN OF TOTALLY SELF-CHECKING ASYNCHRONOUS SEQUENTIAL MACHINES

FUSUN OZGUNER
IBM Watson Research Center
Yorktown Heights, New York

GERNOT METZE
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois

ABSTRACT

State assignments and circuit realizations that lead to totally self-checking asynchronous machine designs are considered. The input, output and state variables are encoded. The effect of each fault on the encoded outputs is studied to obtain a totally self-checking design.

INTRODUCTION

Diagnosing faults in sequential machines continues to be a challenging problem. Since testing by applying test sequences leads to very long sequences even for small machines, it is desirable to explore designs that are easier to test or have self-checking properties.

The design of self-checking asynchronous sequential machines will be discussed in this paper. The circuits considered are assumed to be operating in the fundamental mode, so that a stable state is reached before the inputs are changed. The fault model considered is single stuck faults on gate inputs and outputs.

Definition 1: A circuit is self-testing if, for every fault from a prescribed set, the circuit produces a noncode space output for at least one code space input [1].

Definition 2: A circuit is fault-secure if, for every fault from a prescribed set, the circuit never produces an incorrect code space output for code space inputs [1].

Definition 3: A circuit is totally self-checking if it is both self-testing and fault-secure [1].

Several fault detecting designs have been considered in the literature [2,5, 6], but none is totally self-checking. The approach has been to encode the state variables so that a single fault in the machine causes the machine to become stable in a noncode state, which could be recognized by a fault detecting circuit. However the check circuits used in these designs are not self-checking.

In the design methods presented here, the inputs, outputs and state variables will be encoded. Before discussing specific codes and checkers we will discuss the conditions that the state assignment must satisfy. It has been shown [6] that a fault can cause the machine to become stable in an improper stable state and that the necessary condition to prevent this behavior is to select a state assignment in which the distance between transition paths is at least two units. It has also been shown [3] that in order to obtain monotonic next state equations, each pair of transitions T_i and T_j leading to different stable states in an input column should be partitioned by at least two variables y_i and y_j such that $y_i=1$ and $y_j=0$ for states in T_i and $y_i=0$ and $y_j=1$ for states in T_j.

Definition 4: Controlling variables of a transition are those variables that partition the transition from other transitions in the same input column.

Definition 5: A k-set is a stable state and all the unstable states leading to it in the same input column.

The next state and output equations can be written directly from the flow table and the state assignment, using the controlling variables.

FAULT DETECTING STATE ASSIGNMENTS

A state assignment that meets the conditions stated in the previous section will not necessarily lead to a totally self-checking design. The state assignment code must also be selected in such a way that all gate input and output faults are detected.

A fault detecting design using a 1-out-of-k_p code (where k_p is the number of k-sets in the input column) to encode the k_p-sets of an input column was presented in [5]. It was assumed that the primary input circuitry did not fail. The fault detectors were EX-ORs which checked the parity of the partitioning variables of each input and were also assumed not to fail. Now 1-out-of-k checkers exists for all k except k=3. So, one possible way of obtaining a totally self-checking design would be to use a 1-out-of-k_p checker to check the controlling variables of each input column. Since all the 1-out-of-k_p words are used, the checker will be checked. Then, the outputs of these checkers have to be combined into a global output by two-rail checkers. Of course, the case of having 3 k-sets in an input column, which occurs quite frequently, would have to be treated differently.

In general the k-sets of each input column should be encoded with a constant weight or unordered code, since the conditions stated in Section 1 are met by these codes. For large flow tables the 1-out-of-k_p code requires too many state variables. For example the flow table in Figure 1 has 6 k-sets in input column I. Thus, a 1-out-of-6 code would need 6 variables, whereas a 2-out-of-4 code needs only 4 variables. It can be seen that, for the given state assignment in the 2-out-of-4 code, the distance between any two transition paths is at least two and each transition is partitioned from other transitions by controlling 1's. Therefore each transition can be represented by a product of uncomplemented variables which will be true only during that transition. The product terms representing each k-set are shown in Figure 1. The equations for Y_1 would be in the form

y_1	y_2	y_3	y_4		I
0	0	1	1	......	2
0	0	1	1	......	②
1	1	0	0	......	③
0	1	0	1	......	④
0	1	1	0	......	6
0	1	1	0	......	⑥
1	0	1	0	......	8
1	0	1	0	......	⑧
0	1	0	1	......	4
1	0	0	1	......	⑩

k-sets	state assignment y_1 y_2 y_3 y_4	product terms
(1,2)	0 0 1 1	$I y_3 y_4$
(3)	1 1 0 0	$I y_1 y_2$
(4,9)	0 1 0 1	$I y_2 y_4$
(5,6)	0 1 1 0	$I y_2 y_3$
(7,8)	1 0 1 0	$I y_1 y_3$
(10)	1 0 0 1	$I y_1 y_4$

Figure 1. Example 1. Encoding the k-sets of a large flow table.

$$Y_1 = I\ y_1y_2 + I\ y_1y_3 + I\ y_1y_4 + \ldots.$$

These equations can be written using product terms of uncomplemented variables whenever an unordered code is used to encode the k-sets. The number of product terms is equal to the number of k-sets in the flow table. If shared logic and a two level AND-OR realization is used, an AND gate will realize each product term. This gate will be called the controlling AND gate. Under normal operation only one controlling AND gate will be active for a stable state. Each fault will affect the outputs of these AND gates for some transitions and therefore a 1-out-of-n checker (where n is the number of controlling AND gates) can be used to check the outputs of the controlling AND gates. The checker will be checked when each stable state of the flow table is reached in some way. The fact that each fault affects the outputs of the controlling AND gates is important in analysing the effect of a fault on the outputs. This will be discussed in Section 3.

We now consider the effect of gate faults. A s@0 fault causes the controlling AND gate to go to 0 when it affects a controlling variable. On the other hand, all s@1 faults except those on AND inputs are equivalent to s@1 faults on feedback lines, and a s@1 fault on an AND input or on a feedback line will cause at least two AND outputs to be 1. (For a detailed discussion of the fault-detecting properties of the code see [4]. We will only give an example here.) In Figure 1 suppose that the y_3 input of the $I\ y_3y_4$ gate is s@1. The gate is in fact realizing $I\ y_4$. When the transition controlled by $I\ y_2y_4$ is entered ($y_1y_2y_3y_4 = 0101$), $I\ y_4$ will also be true and since this term appears in the Y_3 and Y_4 equations a stable state with $y_1y_2y_3y_4 = 0111$ will be reached with three AND gates active ($I\ y_3y_4 = 1$, $I\ y_2y_4 = 1$, $I\ y_2y_3 = 1$). Thus, the 1-out-of-n checker will indicate a fault.

The output of the 1-out-of-n checker can be regarded as another output of the circuit.

FAULT DETECTION THROUGH THE OUTPUTS OF THE MACHINE

We will now discuss fault detection through the outputs of the machine, and removing all checkers. In a Mealy machine the output is a function of the input and the state. Therefore the output equations can be written in terms of the same product terms that represent each k-set. The next state and output equations for Example 2 are shown in Figure 2. The state assignment method used is the one used in [6]. However, the outputs are encoded in two-rail form, so that z and z' are complementary under normal operation; zz' = 00 and zz' = 11 are noncode outputs.

It can easily be seen that all faults in the output circuitry cause noncode outputs. A s@0 fault in the two level combinational circuit causes all variables to go to 0 and therefore causes zz' = 00. A s@1 fault affects one or more state variables and will cause two controlling AND gates to be 1 when the input for which the faulty variable is a controlling variable is reached. Again a 1-out-of-n checker could be used to check the outputs of the controlling AND gates, but this is not necessary since the faults

y_1	y_2	y_3	y_4	y_5	y_6	x: 0	1
1	0	0	1	0	0	①,0	2
0	1	0	1	0	0	3	②,0
0	1	0	0	1	0	③,0	4
0	0	1	0	1	0	5	④,0
0	0	1	0	0	1	⑤,0	6
1	0	0	0	0	1	1	⑥,1

(a)

x: 0	1
①,01	2
3	②,01
③,01	4
5	④,01
⑤,01	6
1	⑥,10

(b)

$$Y_1 = x'y_1 + xy_6$$

$$Y_2 = x'y_2 + xy_4$$

$$Y_3 = x'y_3 + xy_5$$

$$Y_4 = x'y_1 + xy_4$$

$$Y_5 = x'y_2 + xy_5$$

$$Y_6 = x'y_3 + xy_6$$

$$z = xy_6$$

$$z' = x'y_1 + x'y_2 + x'y_3 + xy_4 + xy_5$$

(c)

Figure 2. Example 2. (a) Flow table
(b) Flow table with encoded outputs
(c) Design Equations

will eventually affect the outputs. For example, suppose that one of the inputs to the y_5 OR gate in Figure 2 is s@1 and the input is changed to x=1 with the machine in stable state ⑤. Since y_5 is s@1 the state is actually 001 011. When the input changes to x=1, xy_5 and xy_6 will both be 1. From the equations the next state is 101 011 and this state will be reached, because y_5 and y_6 are not changing. But the outputs will go to zz' = 11 because xy_5=1 and xy_6=1, and the fault will therefore be detected. Note, however, that the outputs might not be affected immediately if both product terms appear in the same output equation.

For example suppose that y_3 is s@1 and the transition controlled by $x'y_2$ is entered. Since $x'y_3$ and $x'y_2$ appear in the z' equation the output will be correct. However, if the flow table has no redundant stable states, the fault will eventually cause a noncode output.

Theorem 1: Two stable states of the same input column of a reduced flow table either have different output values, or they produce different output sequences for at least one input sequence. (See [4] for a proof.)

In Example 2 when $x'y_3$ and $x'y_2$ are both true, the state reached is 011 011. When the input is changed to x=1, y_5=1 and y_6=1. Since the output of the stable state of the transition controlled by y_5 =1 is 0 (01 in encoded form) and the output of the stable state of the transition controlled by y_6=1 is 1 (10 in encoded form), the fault will cause the outputs to go to zz'= 11.

In this example all the stable states of the x=0 column have the same output. In general, if a flow table has at least one stable state in each column which has a different output value than the other stable states of that input column, a s@1 fault will cause a noncode output for at least one transition in that input column. Therefore a s@1 fault will be detected through the outputs when all stable states of the flow table are reached.

It should be pointed out that since redundancy cannot be added for hazard protection, inertial delays should be used on the feedback lines to prevent static hazards.

THE (k_p-1)-OUT-OF-k_p CODE

Another code that could be used is the (k_p-1)-out-of-k_p code. This could only be used for small flow tables because it uses a large number of gate inputs. An example is shown in Figure 3. Again a s@0 fault causes all variables to go to 0. A s@1 fault causes all controlling variables of an input column to be 1. If one of the outputs of one of the stable states of the input column is different from the outputs of the other stable states, then a noncode output is reached. If this is not the case a noncode output will be reached eventually. For example in the I_3 column all stable states have the same output value. When a s@1 fault affects a controlling variable the state reached is ①+②+④ (+ denotes bit-by-bit OR) = 1111111. When the input is changed to I_1 or I_2, zz' = 11.

I_1	I_2	I_3	y_1	y_2	y_3	y_4	y_5	y_6	y_7
①, 0	2 , 0	①, 0	0	1	1	1	0	1	0
3 , 0	②, 0	②, 0	1	1	0	1	0	0	1
③, 0	5 , 1	4 , 0	1	1	0	0	1	1	1
1 , 0	5 , 1	④, 0	0	1	1	0	1	1	1
⑤, 1	⑤, 1	4 , 0	1	0	1	0	1	1	1

(a)

$I_1y_2y_3$ ① 011----
$I_1y_1y_2$ ③ 110----
$I_1y_1y_3$ ⑤ 101----

I_2y_4 ② ---10--
I_2y_5 ⑤ ---01--

$I_3y_4y_6$ ① ---1-10
$I_3y_4y_7$ ② ---1-01
$I_3y_6y_7$ ④ ---0-11

(b)

Figure 3. (a) Flow table encoded with a $(k_p=1)$-out-of-k_p code

(b) Controlling variables of each input column and product terms

$$Y_1 = I_1y_1y_2 + I_1y_1y_3 + I_2y_4 + I_2y_5 + I_3y_4y_7$$

$$Y_2 = I_1y_1y_2 + I_1y_2y_3 + I_2y_4 + I_3y_4y_6 + I_3y_4y_7 + I_3y_6y_7$$

$$Y_3 = I_1y_1y_3 + I_1y_2y_3 + I_2y_5 + I_3y_4y_6 + I_3y_6y_7$$

$$Y_4 = I_1y_2y_3 + I_2y_4 + I_3y_4y_6 + I_3y_4y_7$$

$$Y_5 = I_1y_1y_2 + I_1y_1y_3 + I_2y_5 + I_3y_6y_7$$

$$Y_6 = I_1y_1y_2 + I_1y_2y_3 + I_1y_1y_3 + I_2y_5 + I_3y_4y_6 + I_3y_6y_7$$

$$Y_7 = I_1y_1y_2 + I_1y_1y_3 + I_2y_4 + I_2y_5 + I_3y_6y_7 + I_3y_4y_7$$

$$Z = I_1y_1y_3 + I_2y_5$$

$$Z' = I_1y_2y_3 + I_1y_1y_2 + I_2y_4 + I_3y_4y_6 + I_2y_4y_7 + I_3y_6y_7$$

(c)

Figure 3. (Continued)

(c) Design equations

DETECTION OF PRIMARY INPUT FAULTS

Detection of primary input faults was not considered in the literature until now. The next state and output equations will be monotonic in the input variables if the inputs are encoded, for example in the two-rail form. We are assuming that the input code is checked by the output checker of the previous circuit. Therefore the primary input faults that are of interest are those on the inputs of the first level AND gates.

A s@0 fault on the input of an AND gate will be detected when the transition controlled by that AND gate is entered. Since then all variables go to 0 the outputs will go to 0.

A s@1 fault on an AND input is detected in the following way. Consider Example 2. Suppose that the x' input to $x'y_1$ is s@1 and stable state ① controlled by this term is reached. When the input changes to $x=1$, the x' input to $x'y_1$ does not go to 0, because it is s@1. Therefore y_1 remains true. When x goes to 1 xy_4 is also 1, so that y_2 goes to 1. Stable state 110100 is reached, with two controlling AND gates being active. In general this situation will occur when the input is changed to an adjacent input column, after a stable state controlled by the faulty gate is reached, and the fault can be detected by using a 1-out-of-n checker on the outputs of the controlling AND gates. In small flow tables input faults might affect the outputs so that this checker is not needed. However, in larger flow tables a 1-out-of-n checker might be needed to guarantee the detection of these faults.

CONCLUSION

The use of unordered codes to encode the k-sets of each column of an asynchronous flow table to obtain a totally self-checking design was discussed. It was shown that every fault caused a noncode output, so that all single faults could be detected without using a checker. The detection of primary input faults was also discussed.

REFERENCES

[1] Anderson, D.A., "Design of Self-Checking Digital Networks Using Coding Techniques," CSL Report R-527, University of Illinois, September 1971.

[2] Maki, G.K. and Sawin, D.H., "Fault-Tolerant Asynchronous Sequential Machines," IEEE Transactions on Computers, Vol. C-23, pp. 651-657, July 1974.

[3] Mukai, Y. and Tohma, Y., "A Method for the Realization of Fail-Safe Asynchronous Sequential Circuits," IEEE Transactions on Computers, Vol. C-23, pp. 736-739, July 1974.

[4] Ozguner, F., "Design of Totally Self-Checking Asynchronous Sequential Machines," CSL Report R-679, University of Illinois, May 1975.

[5] Patterson, W.W., "On the Design of Diagnosable Asynchronous Sequential Machines," CSL Report R-525, University of Illinois, September 1971.

[6] Sawin, D.H. and Maki, G.K., "Asynchronous Sequential Machines Designed for Fault Detection," IEEE Transactions on Computers, Vol. C-23, pp. 239-249, March 1974.

DIAGNOSIS OF FAULTS IN LINEAR TREE NETWORKS

SHARAD C. SETH
University of Nebraska, Lincoln, Nebraska

K.L. KODANDAPANI
Indian Institute of Technology, Kanpur, India

ABSTRACT

The problem of fault detection and location in tree networks of two input Exclusive-OR (EOR) gates is considered. The fault model assumes that an EOR gate can change to any other function of its two inputs except the equivalence function. An efficient procedure for single fault location is presented. In the worst case the number of tests obtained from the procedure is bounded by a linear function of the number of input variables. Single fault detection in linear trees is extended to the more general class of networks based on the Reed-Muller canonical form. As compared with earlier schemes the proposed extension results in faster networks. Constructive upper bounds are obtained for the number of tests to detect multiple faults in the tree networks. The bounds are all linear functions of the number of input variables.

1. INTRODUCTION

Linear logic networks have been found useful in the encoding and decoding of digital information. They also figure in the functional implementations based on Reed-Muller canonical (RMC) forms [1]. In this paper we study the problems of fault detection and fault location in linear networks.

It is assumed that a linear network is built in the form of a binary tree using the two input Exclusive-OR (EOR) gate as the basic node element. A single fault in the tree corresponds to a failure of any EOR gate to any other function of its two inputs except the Equivalence function*. A multiple fault corresponds to the simultaneous presence of one or more single faults. We will be concerned with the detection and location of single faults (Section 2) and the detection of multiple faults (Section 3). We will also discuss one possible way in which the results on single fault detection could be extended to the more general class of RMC networks (Section 2.3).

2. DIAGNOSIS OF SINGLE FAULTS

2.1 Fault Detection:

It is well known that to detect single faults in linear tree networks four tests are both necessary and sufficient [2]. A procedure for deriving these tests for an arbitrary linear tree network will be presented here. The basic idea is to label each line in the network uniquely with one of three binary vectors: 0011, 0110 or 0101. The labels on the input lines correspond to the required test inputs, while the labels on the other lines indicate the fault free response of the network to these test inputs.

---

*The assumption that no fault can change an EOR gate function to the Equivalence function is valid for all commonly known implementations of the Exclusive-Or function if only stuck type faults are considered. A generalization of this result has been conjectured by several authors: In a single output irredundant combinational network no stuck type fault can result in the complementation of the network function.

Algorithm D (To label a linear tree for single fault detection).

Step D1: Label the output line arbitrarily with one of the three binary vectors: 0011, 0110 or 0101.

Step D2: For each EOR gate whose output line is labelled but whose input lines are not, repeat Step D3 until all lines in the network are labelled.

Step D3: Choosing from the same three vectors label the two input lines of the EOR gate differently from the output line and from each other.

The correctness of the algorithm follows from the following two observations:

(1) Any one of the three binary vectors can be expressed as the component-wise EOR sum of the other two vectors, and

(2) Any pair of the three vectors applied as inputs to an EOR gate tests it exhaustively.

Lemma 1: Let C be the correct response of an EOR tree network to a single fault detection test set obtained from Algorithm D. Then no single fault can change the tree response to $\overline{C}$.

Proof: If a single fault results in the response $\overline{C}$, it is possible only if the output of an EOR gate is itself complemented for all the test inputs. But, since the 4 tests from Algorithm D test each EOR gate in the tree exhaustively, this would imply the gate function changing to the Equivalence function in contradiction of the fault model.

Lemma 1 can be strenghtened to apply to any single fault detection test set because under the assumed fault model it can be easily shown that any test set to detect single faults must apply all four input combinations to each gate.

2.2 Fault Location:

We would like to pinpoint the site of a single fault in a linear tree as accurately as possible; but in as much as a fault on a line connecting two EOR gates in a tree may be considered indistinguishable from certain faults in either of the two gates, it follows that the resolution of faults by external testing can at best be up to two adjacent gates.

The algorithm to be developed for fault location makes use of the principle of binary search. Assume that the network N, viewed as a binary tree, has three components: a left subtree N_L, a right subtree N_R, and the root node R which is the output gate of N. The algorithm will generally try to isolate a fault to one of three components. If R was found to be faulty, the procedure could stop as the fault would be located to a single gate. Otherwise the procedure would be iterated on the faulty subtree.

We will use further notation for brevity. The input pattern of 4 tests obtained by applying Algorithm D to an EOR tree will be denoted by TIP; if an input pattern of all zeros is being applied it will be denoted by ZIP. The symbol C would be used generically to represent the fault free response of an EOR tree to an input pattern. $\overline{C}$ would denote the bit by bit complement of the binary vector C.

Let us consider the effect of applying TIP to N_R and ZIP to N_L. C is the fault free response of N to this pattern of 4 inputs. If no faults are present, C would also be the output of N_R. We will now analyze the possible network responses in the presence of single faults.

If the fault lies in N_L either it does not affect the output of N_L or it affects it in the same way for all the 4 inputs since they remain unchanged. Thus a fault in N_L implies a response of C or $\overline{C}$ for N.

If the output gate R is faulty it is obvious that the response of N can only be one of the four vectors: $\underline{0}$ (all zeros), $\underline{1}$ (all 1's), C, or $\overline{C}$.

A response other than $\underline{0}$, $\underline{1}$, C, or $\overline{C}$ must imply a fault in N_R. In addition a fault in N_R could result in a constant output, but the response C is ruled out because TIP is a single fault detection test set for N_R and the response $\overline{C}$ is ruled out by Lemma 1.

The above analysis applies, mutatis mutandis, to the symmetric case in which TIP is applied to N_L and ZIP is applied to N_R. A summary of the fault isolation obtained from the eight tests thus applied to N is given in Table 1. The "x" entries in the table correspond to logically impossible situations.

TABLE 1

Response of N (first 4 tests)

Response of N (second 4 tests)		$\underline{0},\underline{1}$	$C,\overline{C}$	Other
	$\underline{0},\underline{1}$	R	R,T_L	x
	$C,\overline{C}$	R,T_R	R	T_R
	Other	x	T_L	x

On the other hand the multiple entries such as "R,T_L" in the table appear to belie the hope for a complete fault isolation by this procedure. In such cases, however, further analysis shows (see [7]) that if the procedure is iterated assuming the subtree component was faulty the ambiguous case is guaranteed not to recur. It is thus possible in all cases to isolate a single fault up to two adjacent gates.

The preceding discussion leads to the following algorithm for fault location. It is assumed that fault detection has preceded this algorithm.

Algorithm L (To locate a single fault in a linear tree)

Step L1: While N is a non-trivial tree repeat Steps L2 and L3, otherwise indicate that the fault lies in N and stop.

Step L2: Apply ZIP to N_L and TIP to N_R and record the output. Assume B is the fault free output. If the recorded output is other than $\underline{0}$, $\underline{1}$, B, or $\overline{B}$ replace N by N_R and go back to Step L1. Otherwise continue on to Step L3.

Step L3: Apply ZIP to N_R and TIP to N_L and record the output. Assume C is the fault free output. If the recorded output is other than $\underline{0}$, $\underline{1}$, C or $\overline{C}$ replace N by N_L and go back to Step L1. Otherwise distinguish the following cases on the basis of the responses observed in Steps L2 and L3.

Case 1: (The responses are $[\underline{0},\underline{1}]$ and $[\underline{0},\underline{1}]$ or $[B,\overline{B}]$ and $[C,\overline{C}]$). Indicate gate R to be faulty and stop.

Case 2: (The response is $[\underline{0},\underline{1}]$ and $[C,\overline{C}]$). Replace N by N_R and go back to Step L1.

Case 3: (The response is $[B,\bar{B}]$ and $[0,1]$). Replace N by N_L and go back to Step L1.

Analysis of Algorithm L: The worst case bound on the number of tests is easily obtained. At each iteration at most eight tests are applied and the number of iterations is bounded by (d-1) where d is the depth of the tree. Thus in the worst case 8(d-1) tests will be necessary. For a complete binary tree [3] with m inputs the depth $d = \lceil \log_2 m \rceil$; therefore, in the worst case $8 \lceil \log_2 m \rceil$ tests will be required.

A bound on the average number of tests can be obtained by assuming that all responses to the four tests applied in steps L2 or L3 are equally probable. For any iteration the average number of tests is given by

$$4p + 8(1-p)$$

where p is the probability that the recorded response in Step L3 is not $\underline{0}$,$\underline{1}$, B or $\bar{B}$. Under the assumption of equal likelihood of all responses the above can be rewritten as

$$4.\frac{12}{16} + 8.\frac{4}{16} = 3 + 2 = 5.$$

Thus the average number of tests will not exceed 5(d-1). Again this bound has the smallest value $5 \lceil \log_2 m \rceil$ for an m input complete binary tree.

2.3 Extension to RMC Implementations of Functions:

Reddy [1] has proposed an implementation scheme based on the uncomplemented RMC form of a Boolean function. In this scheme all the Boolean product terms in the RMC form are realized by the first level AND gates and these terms are combined together by a cascade of EOR gates (See Figure 1 for an example). The cascade has an additional variable x_0 which replaces the constant term in the RMC form and which acts as a control input for the purpose of testing the network. This control input plays an important role in enhancing the testability of these networks (hereafter referred to as RMC networks). Reddy has shown that in order to detect a single stuck type

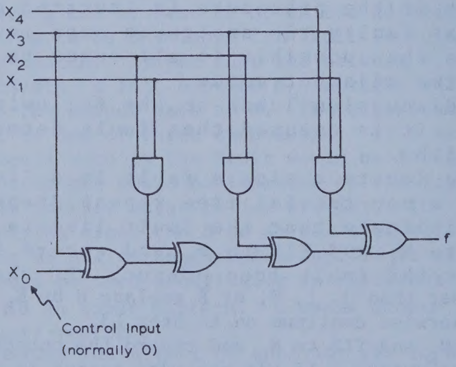


Figure 1: RMC Network for the function $f = x_3 \oplus x_1 x_2 \oplus x_2 x_3 \oplus x_1 x_2 x_3$

fault at the inputs or outputs of AND gates, or an arbitrary fault in a single EOR gate (but not a fault on the primary input bus) it is sufficient to apply only (n+4) tests where n is the number of input variables. Moreover, these tests are derivable independent of the function implementated by the RMC network.

Such ease in testing RMC networks is offset by their two drawbacks: first, since they are based on a canonical form they are often far from minimal in terms of the number of gates, gate inputs, etc.; second, they tend to be very slow because of the cascade of EOR gates.

The extension of RMC networks proposed here tries to alleviate the second drawback by replacing the cascade by a complete binary tree. For n EOR gates in the network this would cut down the delay in the EOR gates by a factor of $n/\lceil \log_2 n \rceil$ as compared to RMC networks.

By Algorithm D we can test any EOR tree with only four tests, but now since the tree inputs are not directly accessible we cannot ensure the application of these four tests without modifying the network. We recall that Algorithm D labels each input to the tree by one of the three binary vectors 0011, 0110, and 0101. In order to obtain these as outputs of the AND gates we propose to add a control input to each AND gate and apply the required vector to it while holding other gate inputs at 1. We note that even though a control input is introduced at every AND gate at most three control variables need to be introduced, one for each distinct label on the tree input. The control variables are labelled as C_1, C_2, and C_3 in Table 2, the first four rows of which test for a single gate fault in the EOR tree. The remaining (n+3) rows in the table test for single stuck at 1 faults at the inputs of the AND gates in a manner discussed in [1]. Other single faults at the AND gates, namely output faults and input stuck at 0 faults, are indistinguishable from certain single faults in the EOR tree, and thus would be detected by the first four tests.

TABLE 2

C_1	C_2	C_3	x_1	x_2	...	x_n
0	0	0	1	1	...	1
0	1	1	1	1	...	1
1	1	0	1	1	...	1
1	0	1	1	1	...	1
0	1	1	1	1	...	1
1	0	1	1	1	...	1
1	1	0	1	1	...	1
1	1	1	0	1	...	1
.	.	.	.	.	...	.
.	.	.	.	.	...	.
1	1	1	1	1	...	0

An examination of Table 2 shows that three of the four tests in the top rows also occur in the bottom n+3 rows. Thus, actually the table represents only (n+4) tests. It is possible to reduce this further to (n+3) tests by the technique discussed in [4].

As an example, consider the function $x_3 \oplus x_1x_2 \oplus x_2x_3 \oplus x_1x_3x_4$. Figure 2 shows the function realized by the extension of RMC networks discussed above.

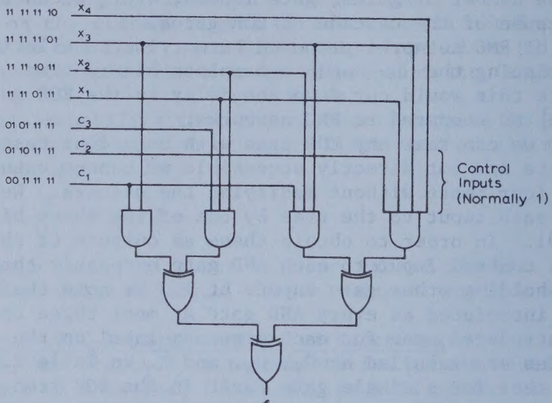


Figure 2

The 8 tests for detecting all single fate faults are marked on the figure. Note that during normal operation the control inputs should be held at the constant value 1.

Unfortunately, no easy extensions of RMC networks using EOR trees appears to be possible for fault location. Algorithm L may, in general, apply arbitrary binary vectors to the tree input which would require not only a control input but also a control variable for each AND gate.

3. DETECTION OF MULTIPLE FAULTS

The ease of detecting single faults in EOR tree networks extends also to multiple faults. In this section we will develop a procedure which allows the derivation of multiple fault detection test sets in an iterative fashion. In the worst case, which occurs when the tree is balanced, multiple faults can be detected by no more than $\left\lfloor \frac{3}{2}n \right\rfloor + 1$ tests where n is the number of inputs to the tree.

Theorem 1: Assume that in an EOR tree N the subtrees N_L and N_R are tested for multiple faults by n_ℓ and n_r tests respectively. Then N can be tested by $n_\ell + n_r$ tests.

Proof: Let T_{N_L} and T_{N_R} be test sets for N_L and N_R of sizes n_ℓ and n_r respectively. The two test sets may be expressed as follows:

$$T_{N_L} = T_{N_L}{}^0 \cup T_{N_L}{}^1 = \{a_1{}^0, a_2{}^0, \ldots, a_j{}^0\} \cup \{a_1{}^1, a_2{}^1, \ldots, a_{n_\ell - j}{}^1\}$$

$$T_{N_R} = T_{N_R}{}^0 \cup T_{N_R}{}^1 = \{b_1{}^0, b_2{}^0, \ldots, b_k{}^0\} \cup \{b_1{}^1, b_2{}^1, \ldots, b_{n_r - k}{}^1\}$$

where the first set on the right hand side in each case includes all the max-terms and the second set includes all the min-terms in the test set. Without loss it can be assumed that $T_{N_L}{}^0$, $T_{N_L}{}^1$, $T_{N_R}{}^0$, and $T_{N_R}{}^1$ are all non-empty.

Define a new set T_N as follows:

$$T_N = \{(x,b_1^0)\ x \in T_{N_L}\} \cup \{a_1^0,y) \mid y \in T_{N_R}\} \cup \{(a_1^1,b_1^1)\}$$

Clearly the size of T_N is $n_\ell + n_r$. We will now prove that T_N detects all multiple faults in N. The proof can be divided into two cases:

Case 1: The multiple fault includes a fault in the output EOR gate.

In such a case we will prove by contradiction that the multiple fault is detected by T_N. Assume that such is not the case and the response of N to T_N matches with the fault free response.

Assume that in the presence of the fault the response of N_L to a_1^0 is a and the response of N_R to b_1^0 is B. Now when $(a_1^0,b_1^0) \in T_N$ is applied to N the output should be zero by the assumption that T_N does not detect the fault. That is, when α is applied to the left input and β to the right input the output gate under fault produces a "0."

Next consider the application of $(a_1^0,b_1^1) \in T_N$ to N. The output of N in this case should be a 1. The left input has to be β ; otherwise the output of N would be 0.

Arguing in a similar way a truth table (Table 3) can be developed for the output gate under fault. This truth table can correspond to only two functions for all possible assignments to α and β: the EOR function and the Equivalence function. Both of these are impossible; the first because R is assumed to be faulty and the second because our fault model does not allow an EOR gate to be complemented.

TABLE 3

Input Applied to N	Left Input of R	Right Input of R	Output of R
(a_1^0,b_1^0)	α	β	0
(a_1^0,b_1^1)	α	β	1
(a_1^1,b_1^0)	α	β	1
(a_1^1,b_1^1)	α	β	0

Case 2: The multiple fault does not include a fault in the output gate.

In this case the argument is symetric with respect to the presence of a fault in either N_L or N_R, therefore, without loss we assume N_L to be faulty. (N_R may or may not be faulty). In order to complete the proof we need the following lemma which is proved in [7].

Lemma 2: Let the correct response of an EOR tree network N to T_N be denoted by C. Then no multiple fault in N can result in a response $\overline{C}$ when T_N is applied to the network.

The proof now goes by contradiction. Assume that the response of N to T_N in the presence of the multiple fault matches the fault free response. In particular, consider the subset $\{(\underline{x},b_1^0) \mid x \in T_{N_L}\}$ of T_N. Since the output of N for this subset is the same as the output for the fault-free function and since the output gate itself is fault-free, there are only two possibilities for the response of N_L to T_{N_L}: (1) it is the same as the fault free response, and (2) it is the complement of the fault free response.

The first possibility leads to the contradiction that N_L is both faulty and fault-free. The second possibility is ruled out by Lemma 2.

Corollary 1: Any n input EOR tree can be tested by 2n tests.

Proof: Let F(n) be the number of tests required to detect multiple faults in an EOR tree with n inputs. Also let n_ℓ and $(n-n_\ell)$ be the number of inputs to test the two subtrees. Then from Theorem 1 we can write the following recursive relation:

$$F(n) = F(n_\ell) + F(n - n_\ell),\ 1<n_\ell <n \tag{1}$$

Also:

$$F(1) = 2 \tag{2}$$

since a line can be tested exhaustively by 2 tests.

It can be easily checked that F(n) = 2n is the unique solution to the equations (1) and (2).

The result in Theorem 1 can be improved further if it is assumed that at least one of the subtrees is non-degenerate (i.e., does not consist of a direct connection from the input to the output.)

Theorem 2:

Assume that at least one of the subtrees N_L and N_R of N is non-degenerate and that N_L and N_R can be tested respectively by n and n_r tests. Then N can be tested by $n_\ell + n_r - 1$ tests.

Proof: See [7].

Corollary 2: Any n input EOR tree can be tested for multiple faults by at most $\lfloor 3n/2 \rfloor + 1$ tests.

Proof: Let the function F have the same meaning as in the proof of Corollary 1. Then the following equations for F can be written:

$$F(n) = F(n_\ell) + F(n-n_\ell) - 1,\quad 1 < n_\ell < n \tag{3}$$
$$F(2) = 4 \tag{4}$$
$$F(1) = 2 \tag{5}$$

It can be easily verified that the solution for F(n) satisfying equations (3), (4), and (5) has a maximum value when the network represents a complete binary tree. In such a case if n is even, only equations (3) and (4) are applicable and it can be seen that $(3/2)n + 1$ is the unique solution. On the other hand, if n is odd equation (5) is used for only one component in the decomposition of N; for all other components, either (3) or (4) are applicable. In this case it can be seen that the solution for F(n) is $(\frac{3(n-1)}{2}+1)+1$. This is equal to $\frac{3n-1}{2} + 1$. In general, the solution can be expressed by the following equation:

$$F(n) = \left\lfloor \frac{3n}{2} \right\rfloor + 1 \tag{6}$$

which encompasses both cases.

The minimum value of F(n) satisfying Equations (3), (4) and (5) occurs when the tree is in the form of a cascade. In such a case Equations (3) and (5) are repeatedly used, Equation (4) is used only once at the end of the decomposition process when the non-degenerate subtree reduces to a single EOR gate (see Figure 3). Thus each input adds an extra test except at the last two inputs add four tests. The solution is clearly F(n) = n + 2. This coincides with the result obtained by Hayes [2].

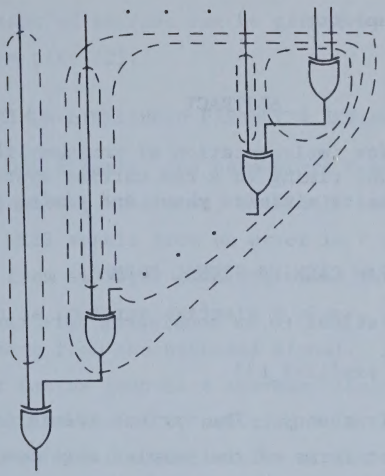


Figure 3: Successive stages in the decomposition of an EOR cascade. At each stage but the last one, one of the components is trivial.

REFERENCES

[1] S.M. Reddy, "Easily testable realizations for logic functions," IEEE Trans. on Computers, Vol. C-21, pp. 1183-1189, Nov. 1972.

[2] J.P. Hayes, "On realizations of Boolean functions requiring a minimal or near-minimal number of tests," IEEE Trans. on Computers, Vol. C-20, pp. 1506-1513, Dec. 1971.

[3] D.E. Knuth, Fundamental Algorithms, Addison-Wesley, Reading, Massachusetts, 1968, pp. 400-401.

[4] K.L. Kodandapani, "A note on easily testable realizations for logic functions," IEEE Trans. on Computers, Vol. C-23, pp. 332-333, March 1974.

[5] M.A. Bruer, "Generation of fault tests for linear logic networks," IEEE Trans. on Computers, Vol. C-21, pp. 79-83, January 1972.

[6] Z. Kohavi, Switching and Finite Automata Theory, McGraw-Hill, New York, 1970, P. 104.

[7] S.C. Seth and K.L. Kodandapani, "Diagnosis of Faults in Linear Tree Network," submitted to IEEE-C (available through Repository)

PARAMETER ESTIMATION FOR DIGITAL DATA SIGNALS

L. E. FRANKS
University of Massachusetts
Amherst, Mass.

ABSTRACT

Receiver structures for implementation of maximum-likelihood estimation of carrier phase and symbol timing in a PAM carrier system are presented. The design of pulse shapes to minimize phase and timing jitter is considered.

PAM CARRIER-SIGNAL FORMATS

The types of carrier signal to be considered here can be expressed as

$$y(t) = \mathrm{Re}[\beta(t)\ \exp\{j2\pi f_o t\}] \tag{1}$$

where f_o is the carrier frequency. The various modulation formats are characterized by different forms of the complex envelope, $\beta(t)$; however, the carrier phase parameter, θ, and the symbol timing parameter, τ, influence the complex envelope the same way in all cases. For double-sideband pulse amplitude modulation (DSB-PAM) carrying a data sequence, $\{a_k\}$, at a symbol rate of 1/T we have:

$$\begin{aligned} \beta(t;\ \tau,\ \theta) &= \sum_k a_k\ \gamma(t-kT-\tau)\ e^{j\theta} \\ \text{where } \gamma(t) &= g(t) + j0 \end{aligned} \tag{2}$$

For vestigial-sideband modulation (VSB-PAM), the format is the same as in (2) except that

$$\begin{aligned} \gamma(t) &= g(t) + j\ \tilde{g}(t) \\ \text{where } \tilde{G}(f) &= F(f)\ G(f) \end{aligned} \tag{3}$$

The quadrature pulse, $\tilde{g}(t)$, can be generated by a transfer function, F(f), which is equal to j over the baseband frequency region corresponding to complete suppression of one of the sidebands. In the extreme case of single-sideband modulation (SSB-PAM) we have $F(f) = -j\ \mathrm{sgn}\ f$ and $\tilde{g}(t) = \hat{g}(t)$, the Hilbert transform of g(t).

Another modulation format, called quadrature amplitude modulation (QAM), carries two data sequences simultaneously in the following manner:

$$\beta(t;\tau,\ \theta) = e^{j\theta} \sum_k a_k\ g(t-kT-\tau) + jb_k\ h(t-kT-\tau) \tag{4}$$

This work was supported by the Air Force Office of Scientific Research, Air Force Systems Command, under Grant No. AFOSR-71-2111.

where g(t) and h(t) are real baseband pulses. In normal QAM, the same pulse shapes are used in the in-phase and quadrature channels, $h(t) = g(t)$; however, some important advantages can be gained by using staggered QAM (SQAM) wherein $h(t) = g(t-T/2)$.

MAXIMUM-LIKELIHOOD PARAMETER ESTIMATION

In the VSB and QAM formats, the parameters θ and τ must be accurately determined at the receiver; otherwise, crosstalk between the in-phase and quadrature channels will result from an error in θ and intersymbol interference will result from improper sampling times due to an error in τ. A traditional approach is to first estimate θ alone, perform the demodulation, and then recover timing from the baseband signal. It is known that a version of the familiar Costas loop is a maximum-likelihood estimator for θ.[1] Maximum-likelihood estimation of τ, assuming θ is correct, can be implemented by operations on the time-derivative of the squared output of a baseband matched filter. An equivalent implementation [2] is shown in Fig. 1. Estimation of the data sequence can then be performed on the output samples of the matched filter [3]. This approach is satisfactory for DSB-PAM, however the phase estimation fails under normal conditions of VSB and QAM [1].

An alternative approach which has received considerable attention recently [4]-[9] is called "data-aided" recovery. This involves joint estimation of τ, θ, and the data sequence, $\{a_k\}$. Implementation involves tracking loops similar to those in Fig. 1, but they are all interacting and are controlled by the estimated data sequence. This causes some implementation difficulties if the sequence estimator introduces excessive time delay. This motivates our present investigation into the possibility of joint estimation of τ and θ only, without regard to the data sequence.

Assuming that the received signal is $y(t) + u(t) = \mathrm{Re}[\alpha(t)\exp\{j2\pi f_o t\}]$ where u(t) is Gaussian noise with a flat spectral density of N_o watts/Hz, the likelihood function for joint estimation of τ, θ, and the data sequence is given by [4], [5]

$$L(\tau, \theta) = C \exp\left[\frac{1}{2N_o} \mathrm{Re}(\alpha, \beta) - \frac{1}{4N_o} (\beta, \beta)\right]$$
$$\text{where } (\alpha, \beta) \triangleq \int_{T_o} \alpha(t)\, \beta^*(t)\, dt \qquad (5)$$

For convenience, we have assumed that the true values of τ and θ are zero, hence in (5) and the following, τ and θ denote the estimates of these quantities. The appropriate likelihood function for τ and θ, independent of the data sequence, is obtained by averaging $L(\tau, \theta)$ in (5) over the data sequence. Assuming the $\{a_k\}$ are independent, identically-distributed Gaussian random variables an approximate expression for the log-likelihood function, omitting unnecessary constants, can be expressed for the VSB case as

$$\Lambda_o(\tau, \theta) = \ell n \underset{a_k}{E} [L(\tau, \theta)] = \sum_k q_k^2 \tag{6}$$

where we let

$$q_k \triangleq \mathrm{Re}(\alpha, \gamma_k(\tau)\ e^{j\theta}) \tag{7}$$

and $\gamma_k(\tau)$ denotes the complex baseband pulse, γ, in (2) and (3) shifted in time by an amount $kT + \tau$. The quantity, q_k, can be interpreted as the response of a bandpass filter matched to the received signal pulse, demodulated against a reference phase, θ, and sampled at $t = kT + \tau$, as shown in Fig. 2. With this interpretation, the index, k, in (6) is assumed to run only over $K = T_o/T$ values.

One possible receiver implementation, shown in Fig. 3, seeks a maximization of (6) by using partial derivatives of $\Lambda_o(\tau, \theta)$ with respect to τ and θ as control signals to drive a voltage-controlled oscillator (VCO) and a voltage-controlled clock (VCC) in a closed-loop tracking arrangement. If, after several successive observation intervals, the oscillator and clock phases converge to the correct value, we would have the condition

$$\begin{aligned} w_1(\theta) &\triangleq \left.\frac{\partial \Lambda_o}{\partial \theta}\right|_{\tau=0} = 0 \text{ at } \theta = 0. \\ w_2(\tau) &\triangleq \left.\frac{\partial \Lambda_o}{\partial \tau}\right|_{\theta=0} = 0 \text{ at } \tau = 0. \end{aligned} \tag{8}$$

Since the received signal, $\alpha(t)$, contains a noise term and another term depending on the random data sequence, the zero-crossings of the "timing waves", w_1 and w_2, do not occur exactly at the "true" values, $\theta=0$ and $\tau=0$. The result is a random phase jitter and timing jitter in the estimates. At moderately high signal-to-noise ratio the jitter due to noise is negligible compared to that produced by the data sequence. We can examine the data interference in the recovery process by evaluating $w_1(\theta)$ and $w_2(\tau)$ for $\alpha(t) = \beta(t; 0, 0)$. The dependence of the timing waves on the baseband pulse shape is of particular interest. It has already been noted [10], [11] that proper prefiltering of a baseband PAM signal before it enters the timing recovery circuit can lead to jitter-free recovery in the noise-free case.

PULSE SHAPING TO MINIMIZE JITTER

For the VSB (SSB) case, we find that

$$w_1(\theta) = \sum_{k,\ell,m} a_\ell a_m \, \mathrm{Re}[-je^{-j2\theta} s(kT-mT)s(kT-\ell T) + j\, s(kT-mT)s^*(kT-\ell T)] \tag{9}$$

where

$$s(t) = 2r(t) + j2\hat{r}(t) \; ; \; R(f) = |G(f)|^2 \tag{10}$$

In most cases, we are interested in baseband pulses whose bandwidth is not greatly in excess of the Nyquist bandwidth, 1/2T. It is convenient to express $r(t)$ as

$$r(t) = \mathrm{Re}[c(t)e^{j\frac{\pi t}{T}}] \tag{11}$$

where $C(f)$ is real to satisfy (10) and the bandwidth of $C(f)$ does not exceed 1/2T. Choosing $c(t)$ real makes $\hat{r}(mT) = 0$ for all m and $w_1(0) = 0$, independently of the data sequence $\{a_k\}$. Hence we have jitter-free phase recovery (provided that τ is correct) if $R(f)$ has even symmetry about 1/2T. On the other hand, we find

$$w_2(\tau) = 2 \sum_{k,\ell,m} a_\ell a_m \; r(kT-mT+\tau) \; \dot{r}(kT-\ell T+\tau). \tag{12}$$

and jitter-free timing recovery can be obtained with $c(t) = a(t)+j(T/\pi)\dot{a}(t)$ where $a(t)$ is real. It should be noted that these are only sufficient conditions on $c(t)$ for elimination of jitter and other designs might lead to a smaller combined jitter. Unfortunately, it appears impossible to eliminate both types of jitter simultaneously with pulses with bandwidth less than twice the Nyquist bandwidth.

Turning now to the QAM case, the log-likelihood function (6) is modified to

$$\Lambda_o(\tau, \theta) = \sum_k p_k^2 + q_k^2 \tag{13}$$

$$p_k \triangleq \mathrm{Re}(\alpha, \; g_k(\tau) \; e^{j\theta}) \; ; \; q_k \triangleq \mathrm{Re}(\alpha, \; j \; h_k(\tau) \; e^{j\theta})$$

Evaluating the corresponding $w_1(\theta)$, we find that it vanishes for all θ in the normal (g=h) case, hence phase recovery cannot be obtained. Fortunately, SQAM gives an entirely different result. If we shape pulses so that $r(mT \pm T/2) = 0$ for all m, which can be done by making $c(t)$ real in (11), then

$$w_1(\theta) = \sum_{k,\ell,m} [a_\ell a_m + b_\ell b_m] \; \mathrm{Re}[-je^{-j2\theta} r(kT-mT)r(kT-\ell T)] \tag{14}$$

and jitter-free phase recovery results (provided τ is correct). The other timing wave, $w_2(\tau)$, is identical to (12) except that $a_\ell a_m$ is replaced by $a_\ell a_m + b_\ell b_m$.

A method for evaluating r.m.s. values of jitter which appears promising for a more complete study of this problem is one developed recently[10] for analysis of baseband timing recovery. The method is based on the fact that $w_1(\theta)$ and $w_2(\tau)$ are cyclostationary random processes and general techniques have been developed [12] for analysis of the second-order statistical properties of this class of processes. To illustrate the method, consider a timing wave, $w_2(\tau)$, whose zero-crossings tend to be tightly clustered about the correct point, Then we can write

$$\dot{w}_2(t_o)\tau \cong -w_2(t_o) \tag{15}$$

where t_o is the zero-crossing point of the mean timing wave, $E[w_2(\tau)]$. An appropriate expression for r.m.s. jitter is obtained from

$$E[\tau^2] \cong E[w_2^2(t_o)]\{E[\dot{w}_2(t_o)]\}^{-2} \tag{16}$$

Since $w_2(\tau)$ is cyclostationary, its mean value and mean-squared value are periodic functions of τ. Furthermore, with sharply bandlimited pulses, the Fourier series expansions of these functions typically contain only one or two terms. For example, with $w_2(\tau)$ as given in (12) and assuming independent, identically-distributed, zero-mean, unit-variance, Gaussian data, we find that

$$\begin{aligned} E[w_2(\tau)] &= \frac{\pi}{T^2} A \sin\frac{2\pi\tau}{T} \\ \mathrm{Var}[w_2(\tau)] &= \frac{\pi^2 A^2}{T^2}\left[1 - \cos\frac{4\pi\tau}{T}\right] + \frac{AB}{T^2}\left[1 + \cos\frac{2\pi\tau}{T}\right]^2 \end{aligned} \tag{17}$$

where

$$\begin{aligned} A &\triangleq \int_{-\infty}^{\infty} C^2(f)\,df \\ B &\triangleq \int_{-\infty}^{\infty} (2\pi f)^2 C^2(f)\,df \end{aligned} \tag{18}$$

and from these expressions it is a relatively simple matter to evaluate the r.m.s. value of τ using (16).

REFERENCES

1. L. E. Franks, "Acquisition of Carrier and Timing Data-I." Proceedings NATO Advanced Study Institute on New Directions in Signal Processing in Communication and Control, Noordhoff. Int'l Publ. Co., 1975.

2. R. D. Gitlin and J. Salz, "Timing Recovery in PAM Systems," Bell Syst. Tech. Jour., vol. 50, pp. 1645-1669, May-June 1971.

3. G. D. Forney, Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," IEEE Trans., vol. IT-18, pp. 363-378, May 1972.

4. H. Kobayashi, "Simultaneous Adaptive Estimation and Decision Algorithm for Carrier Modulated Data Transmission Systems," IEEE Trans., vol. COM-19, pp. 268-280, June 1971.

5. G. Ungerboeck, "Adaptive Maximum-Likelihood Receiver for Carrier-Modulated Data Transmission Systems," IEEE Trans., vol. COM-22, pp. 624-636, May 1974.

6. W. C. Lindsey and M. K. Simon, "Data-aided Carrier Tracking Loop," IEEE Trans., vol. COM-19, pp. 157-168, April 1971.

7. M. K. Simon and J. G. Smith, "Carrier Synchronization and Detection of QASK Signal Sets," IEEE Trans., vol. COM-22, pp. 98-106, February 1974.

8. __________, "Offset Quadrature Communications with Decision-Feedback Carrier Synchronization," IEEE Trans., vol. COM-22, pp. 1576-1584, October 1974.

9. R. Matyas and P. J. McLane, "Decision-Aided Tracking Loops for Channels with Phase Jitter and Intersymbol Interference," IEEE Trans., vol. COM-22, pp. 1014-1023, August 1974.

10. L. E. Franks and J. P. Bubrouski, "Statistical Properties of Timing Jitter in a PAM Timing Recovery Scheme," IEEE Trans., vol. COM-22, pp. 913-920, July 1974.

11. E. Roza, "Analysis of Phase-Locked Timing Extraction Circuits for Pulse Code Transmission," IEEE Trans., vol. COM-22, pp. 1236-1246, September 1974.

12. W. A. Gardner and L. E. Franks, "Characterization of Cyclostationary Random Signal Processes," IEEE Trans., vol. IT-21, pp. 4-14, January 1975.

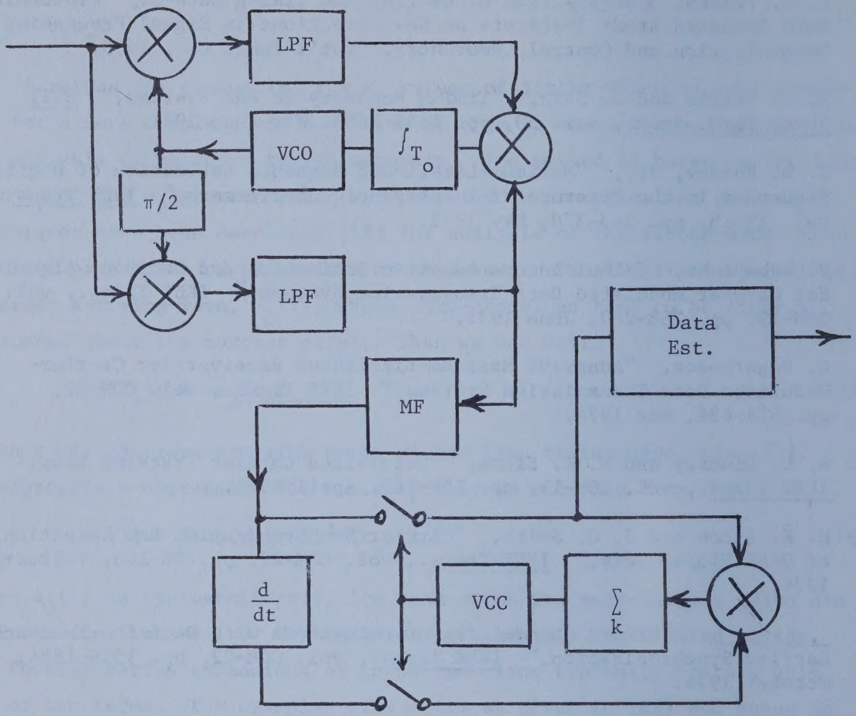

LPF
VCO
$\int_{T_o}$
$\pi/2$
LPF
Data Est.
MF
$\frac{d}{dt}$
VCC
$\sum_k$

Figure 1: DSB-PAM Receiver. MF denotes a filter matched to the baseband pulse.

$$q_k(\tau,\theta) = \mathrm{Re}(\alpha,\gamma_k(\tau)e^{j\theta}) \cong \mathrm{Re}[e^{-j\theta}\int_{-\infty}^{\infty} \alpha(\sigma)\gamma^*(\sigma-kT-\tau)\,d\sigma]$$

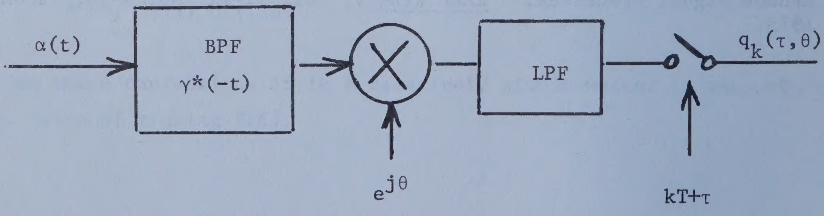

$\alpha(t)$
BPF
$\gamma^*(-t)$
LPF
$q_k(\tau,\theta)$
$e^{j\theta}$
$kT+\tau$

Figure 2: Evaluation of q_k.

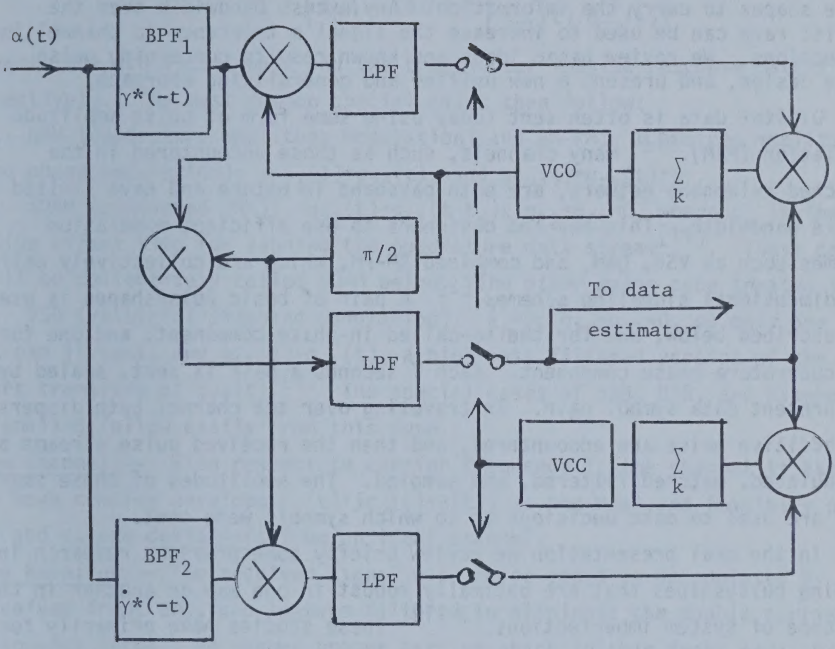

α(t)
BPF$_1$
γ*(-t)
LPF
VCO
∑
k
π/2
To data
estimator
LPF
VCC
∑
k
BPF$_2$
γ̇*(-t)
LPF

Figure 3: Joint carrier phase and symbol timing recovery.

ON OPTIMAL PULSE-SHAPE DESIGN FOR DATA TRANSMISSION

FRANCIS S. HILL, JR.
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, Massachusetts 01002

Abstract

Linear modulation schemes (PAM-VSB, QAM, AM-PM, etc.) utilize Nyquist pulse shapes to carry the information. Any excess bandwidth over the Nyquist rate can be used to increase the signal's tolerance to channel imperfections. We review basic ideas and known results concerning pulse shape design, and present a new unified and generalizing approach.

Digital data is often sent today using some form of pulse amplitude modulation (PAM).[1] Many channels, such as those encountered in the switched telephone network, are both passband in nature and have limited usable bandwidth. This has led designers to use efficient modulation schemes such as VSB, QAM, and combined AM-PM, which are collectively called two-dimensional signaling schemes.[2] A pair of basic pulseshapes is used, as described below, one for the so-called in-phase component, and one for the quadrature-phase component. Each T seconds a pair is sent, scaled by the present data symbol pair. In traveling over the channel both dispersion and additive noise are encountered, and then the received pulse streams are demodulated, matched filtered, and sampled. The amplitudes of these samples are used to make decisions as to which symbols were sent.

In the oral presentation we review briefly some previous research in finding pulseshapes that are optimally robust in one way or another in the presence of system imperfections.[3-11] These studies have primarily focused on sampling time and carrier phase jitter, and have usually dealt with a specific signaling scheme such as SSB.

Here the pulseshape design problem is examined in a unified two-dimensional signaling context, and several new results are presented. Only a skeleton of the derivations is given. A forthcoming publication will provide more leisurely discussion.[12] The basic problem is to find bandlimited pulseshapes which are simultaneously Nyquist (i.e. intersymbol interference-free) and optimally tolerant to small channel deviations from an ideal characteristic. The constraint of being Nyquist leaves design freedom only if the pulse bandwidth exceeds the Nyquist bandwidth;[1] hence the design problem here relates to optimal utilization of whatever excess bandwidth is available.

To establish a framework we slightly modify the approach of Falconer and Foschini[13] and use the system model of Figure 1, which is general enough to include any two-dimension linear modulation scheme. Two data streams of symbols are output from the source, one pair each T seconds. They modulate four transmitter filters, and are heterodyned by the two carrier sinusoids in quadrature to form the transmitted signal:

$$s_1(t) = \begin{Bmatrix}\cos(2\pi f_0 t)\\ \sin(2\pi f_0 t)\end{Bmatrix}' \sum_n g(t-nT)\underset{\sim}{a}_n \qquad (1)^*$$

where

$$\underset{\sim}{a}_n = \begin{Bmatrix} a_n \\ \check{a}_n \end{Bmatrix}, \quad g(t) = \begin{Bmatrix} g_{11}(t) & g_{12}(t) \\ g_{21}(t) & g_{22}(t) \end{Bmatrix} \qquad (2)$$

denote the nth symbol pair and the transmitter impulse response matrix respectively. The most common special cases then follow:

1. QAM (quadrature amplitude modulation) and AM-PM: (Combined amplitude and phase modulation): $g_{11}(t)=g_{22}(t)$, and $g_{12}(t)=g_{21}(t)=0$.
2. SQAM (staggered QAM): $g_{22}(t)=g_{11}(t-t_o)$; $g_{12}=g_{21}=0$, where t_o is the relative offset time for sending the quadrature data stream[11]. These cases will be collectively called SQAM below. The other major case treated is:
3. VSB (vestigial sideband modulation): $\check{a}_n=0$; $g_{12}=g_{22}=0$, so only one data stream is sent, and $g_{21}(t)=\tilde{g}_{11}(t)$, a high-pass filtered version of the Hilbert transform of $g_{11}(t)$.[9] The special cases of SSB, DSB, and baseband signaling follow easily from this type.

The Channel — With respect to carrier frequency f_o the channel is assumed to have complex envelope $c_1(t)+jc_2(t)+\delta(t)$, so the real and imaginary parts c_1 and c_2 are deviations from an ideal channel.

The Receiver — The received signal with noise added is demodulated by the receiver front end, and lowpass filtered to eliminate the double carrier frequency terms. We assume proper carrier phase in this front end: Later any actual phase error will be treated as an imperfection in the channel. Note the factor λ in the lower path. λ has value 1 for SQAM signaling, but for VSB schemes $\lambda = 0$, since this path is not used.

The front end is followed by a matrix matched filter.[13] Because the channel deviations to be treated will be both small and unknown, it is reasonable that these filters are matched to the transmitted pulse shapes—as if the channel were ideal—so they have impulse responses $g_{ij}(-t)$. As the positions of g_{12} and g_{21} are reversed at the receiver, the transpose of matrix g(-t) is used. The output pair of signals is then sampled each T seconds. Due to assumed stationarity of data and noise processes it is sufficient to analyze the properties of the single sample pair y(0) taken at t=o, assuming that decisions are made on a symbol-by-symbol basis. The sampling time is assumed perfect here, as any timing error will be treated later as a channel imperfection. Since the voltage y(0) ideally has value $\underline{a}_0$, the error voltage is $\underline{y}(0)-\underline{a}_0$. We measure performance using a mean squared error criterion based on the sum of the mean squared errors of the in-phase and quadrature terms. It is most revealing to express MSE in frequency domain terms. After standard but lengthy manipulations one obtains

---

*A glossary of symbols appears at the end of the paper.

$$MSE = \frac{1}{T}\, tr \int_{-1/2T}^{1/2T} df\ (rep\ H(f))m(f)(rep\ H(f))^{*\prime}+N \tag{3}$$

where $H(f)=G'(f)KC^*(f)G^*(f)$ is the deviation of the overall system transfer function from the ideal, N is the trace of the noise sample covariance matrix, rep,tr,',*, etc. are defined in the Glossary below, G(f) is the term-by-term Fourier transform of matrix g(t),C(f) is the channel deviation matrix

$$C(f) = \begin{Bmatrix} C_1(f) & C_2(f) \\ -C_2(f) & C_1(f) \end{Bmatrix}, \quad K = \begin{pmatrix} 1 & 0 \\ 0 & \lambda \end{pmatrix} \tag{4}$$

and M(f) captures the correlation properties of the data symbols:

$$M(f) = \sum_k E[\underline{a}_{n+k}\, \underline{a}_n{}'] \, e^{j2\pi fkT} = \begin{pmatrix} m_{11}(f) & m_{12}(f) \\ m_{21}(f) & m_{22}(f) \end{pmatrix}. \tag{5}$$

It is easy to show that $m_{21}(f)=m_{12}^*(f)$.

At this point we wish to display more explicitly the dependence of MSE on the pulse shapes in G(f), for the two basic cases SQAM and VSB.

1. SQAM: From the discussion following (2), $G(f)=G_1(f)\begin{pmatrix} 1 & 0 \\ 0 & x \end{pmatrix}$, where $G_1(f)$ is a scalar function and $x=\exp(-j2\pi ft_o)$ represents the effect of staggering. H(f) in (3) then depends on $|G_1(f)|^2$, the real and positive pulse shape to be designed. As it must be Nyquist it has the form[8]: $|G_1(f)|^2=T\cdot[P(f)+B(f-1/2T)-B(f+1/2T)]$, where P(f) is of rectangular shape and B(f) is a real and odd function that specifies the rolloff shape as in Figure 2a. We assume an allowable excess bandwidth of $\gamma/2T$ Hz, with $\gamma\leq 1$. The design problem is to find optimal shapes B(f). Using this form in (3), and making for simplicity the usual assumptions that the two data streams are statistically identical and uncorrelated ($m_{22}=m_{11}$ and $m_{12}=0$), (3) becomes for SQAM:

$$MSE_Q = 2T\int_{-1/2T}^{1/2T} df\ m_{11}(f)(|a|^2+|b|^2)+2T\int_{0}^{\gamma/2T} df\ m_{11}(f+1/2T)\{2B(f)W_Q(f)+B^2(f)U_Q(f)\}+N \tag{6}$$

where

$$W_Q(f) = 2\ Re[a_-(a_+-a_-)^* + b_+^*b_-\ \cos(2p)-|b_-|^2] \tag{7}$$

$$U_Q(f) = 2|a_+-a_-|^2+|b_+-b_-e^{j2p}|^2+|b_+-b_-e^{-j2p}|^2 \tag{8}$$

and $p=\pi t_o/T$, $a=C_1(f)$, $b=C_2(f)$, and $a_+=C_1(f+1/2T)$, $a_-=C_1(f-1/2T)$, etc, indicate translated version of the channel functions.

2. VSB: Here we obtain:

$$G(f) = \begin{pmatrix} G_1(f) & 0 \\ -jV(f)G_1(f)\ \mathrm{sgn}(f) & 0 \end{pmatrix} \tag{9}$$

where V(f) is a highpass characteristic that determines the vestigial roll-off band$^{(9)}$, as suggested in Figure 2b). V(f) is unity for $|f| \geq \beta/2T$, allowing a vestigial band of excess bandwidth $\beta/2T$. Again using the relation between $|G_1|^2$ and B(f), tedious manipulations yield:

$$MSE_V = T\int_{-1/2T}^{1/2T} df\ m_{11}(f)|a - jbV^*(f)\mathrm{sgn}(f)|^2 + 2T\int_0^{\gamma/2T} df\ m_{11}(f+1/2T)[2B(f)W_V(f) - B^2(f)U_V(f)] + N \tag{10}$$

$$\begin{aligned} \text{where } W_V(f) &= \mathrm{Re}[\{(a_+ - a_-) - j(b_+ + b_-)\}(a_- + jb_-)^*] \\ U_V(f) &= |(a_+ - a_-) - j(b_+ + b_-)|^2 \quad . \end{aligned} \tag{11}$$

Equations (6), (7) and (10) permit the evaluation of MSE due to intersymbol interference, cross talk, and noise for a given channel, if all the ingredients are known. In addition, optimum transmitter pulse shapes follow by classical variational techniques $^{(14)}$: For SQAM: $B_{opt}(f) = -W_Q(f)/U_Q(f)$, while for VSB: $B_{opt}(f) = -W_V(f)/U_V(f)$ and $V_{opt}(f) = ja^*(f)\mathrm{sgn}(f)/b^*(f)$. Note that if b(f)=0 (no channel crosstalk) then the shape of $V(\cdot)$ is irrelevant to performance. If a(f)=0, on the other hand, then by inspection V(f) should be zero over the allowed vestigial band.

To proceed we assume the channel deviation functions are parameterized by vector $\underline{\theta}$ of parameters specifying the 'amount' of each kind of distortion with the proviso that as $\underline{\theta}$ vanishes so do the channel functions: $C_i(f) = C_i(f,\underline{\theta})$ such that $C_i(f,\underline{0})=0, i=1,2$. For example, if the channel introduces quadratic delay distortion of amount θ, centered at f_1 Hz, the lowpass deviation is $C(f,\theta) = \exp(j\theta(f+g)^3) - 1$ where $g = f_0 - f_1$. Now $C_1(f,\theta)$ is the transform of the real part of $c(t,\theta)$, found from the symmetrical part of $C(f,\theta)$:$^{(14)}$ $C_1(f,\theta) = 1/2(C(f,\theta) + C^*(-f,\theta))$, and similarly for C_2. Hence for this example

$$\begin{aligned} C_1(f,\theta) &= \exp(j\theta(f^3+3fg^2))\cos(\theta(3f^2g+g^3)) - 1 \\ C_2(f,\theta) &= \exp(j\theta(f^3+3fg^2))\sin(\theta(3f^2g+g^3)) \quad . \end{aligned} \tag{12}$$

Under the likely conditions that the exact value of $\underline{\theta}$ will not be known one could neither evaluate MSE nor design optimal shapes $B(\cdot)$ and $V(\cdot)$. One could instead attempt to minimize MSE averaged over all channel uncertainty. From (6) (7) and (8) one then obtains for SQAM the optimum shapes $B(f) = -E_{\underline{\theta}}W_Q(f)/E_{\underline{\theta}}U_Q(f)$, and similarly for VSB. However, even if the joint pdf of the vector $\underline{\theta}$ is known, these averaging operations may be very difficult to perform.

A simpler situation prevails if we assume the parameter values $\underline{\theta}$ are small. Then MSE ($\underline{\theta}$) may be expanded in a Taylor series about $\underline{\theta} = \underline{0}$:

$$\mathrm{MSE}(\underline{\theta}) = N + 1/2\ \underline{\theta}^T H \underline{\theta} + \cdots \tag{13}$$

where H is the Hessian matrix with i-jth element $H_{ij} = \partial^2 \mathrm{MSE}(\theta)/\partial\theta_i \partial\theta_j$ evaluated at $\underline{\theta} = \underline{0}$. The problem is then to select shapes $B(\cdot)$ and $V(\cdot)$ to minimize some functional of the matrix H. If $\underline{\theta}$ is random then from (13) MSE averaged over $\underline{\theta}$ involves only second moments of the unknown parameters, which is much simpler to deal with than the entire MSE functional. If only one parameter is involved, one would attempt to minimize the scalar H_{11}. If several are involved it is revealing to assume[8] that some linear relation $\underline{\theta} = L\theta_1$ between the θ_i holds, where L is a column vector of fixed numbers. Then one would minimize L'HL. In any case the elements H_{ij} follow easily by partial differentiation of (6) (8) and (10). For simplicity we pursue the single parameter case here.

Single Parameter Case — Taking second partials in (7) and (8) yields for the optimum B(f) for SQAM:

$$B(f) = -\ddot{W}_Q(f)/\ddot{U}_Q(f) \tag{14}$$

where

$$\ddot{W}_Q(f) = 4\ \mathrm{Re}[\dot{a}_-(\dot{a}_+-\dot{a}_-)^* + \cos(2p)\dot{b}_+^*\dot{b}_- - |\dot{b}_-|^2] \tag{15}$$

$$\ddot{U}_Q(f) = 2[2|\dot{a}_+-\dot{a}_-|^2 + |\dot{b}_+-\dot{b}_- e^{j2p}|^2 + |\dot{b}_+-\dot{b}_- e^{-j2p}|^2 \tag{16}$$

and $\dot{a}$, etc. denote partials evaluated at $\theta=0$. Note that the optimum $B(\cdot)$ does not depend on the data correlation $m_{11}(f)$, nor on the amount of excess bandwidth γ. B(f) is understood to be zero beyond $\gamma/2T$. Now put B(f) back into H_{11} to obtain its minimum value:

$$H_{Q\,\min} = 4T \int_{-1/2T}^{1/2T} df\ m_{11}(f)(|\dot{a}|^2 + |\dot{b}|^2) + 2T\int_0^{\gamma/2T} df\ m_{11}(f+1/2T)\ \ddot{W}_Q(f)B(f)\ . \tag{17}$$

Similarly for VSB, taking partials in (10) yields

$$B(f) = -\ddot{W}_v(f)/\ddot{U}_v(f)\quad ,\ V(f) = j\dot{a}^*(f)\mathrm{sgn}(f)/\dot{b}^*(f) \tag{18}$$

$$\ddot{W}_v(f) = 2\ \mathrm{Re}[\{(\dot{a}_+-\dot{a}_-)^* + j(\dot{b}_++\dot{b}_-)^*\}(\dot{a}_-+j\dot{b}_-)] \tag{19}$$

$$\ddot{U}_v(f) = 2|(\dot{a}_+-\dot{a}_-)-j(\dot{b}_++\dot{b}_-)|^2 \tag{20}$$

and from (10) the minimum H_{11} is:

$$H_v\ \min = 2T\int_{-1/2T}^{1/2T} df\ m_{11}(f)|\dot{a}^* + jV(f)\dot{b}^*\mathrm{sgn}(f)|^2 + 2T\int_0^{\gamma/2T} df\ m_{11}(f+1/2T)\ddot{W}_v(f)B(f). \tag{21}$$

If $\dot{b}(f)=0$ the first integrand is just $m_{11}(f)|\dot{a}(f)|^2$, while if $\dot{b}(f)\neq 0$ it vanishes for $|f|<\beta/2T$ and is $m_{11}(f)|\dot{a}^*+j\dot{b}^*\mathrm{sgn}(f)|^2$ elsewhere. If $\dot{b}=0$ then $\ddot{W}_v = 1/2\ddot{W}_Q$ and $\ddot{U}_v = 1/2\ddot{U}_Q$, so B(f) is formally the same for all signaling cases.

Comparison of VSB and SQAM — If each source symbol in Figure 1 can take on one of M values, it is clear that for equal rates one must send VSB pulses twice as often, so $T_V=1/2T_Q$. Assuming a total usable passband bandwidth of W_1 on the channel, then for VSB, $W_1=(1+\gamma_V+\beta_V)/2T_V$, while for SQAM $W_1=(1+\gamma_Q)/T_Q$. Equal rates and bandwidths then force

$$\gamma_Q=\gamma_V+\beta_V \text{ and } T_Q=2T_V, \text{ and } N_Q=2N_V, \tag{21}$$

since there is only one noise path in VSB signaling. In calculations of MSE to follow these relations will be followed. There is still the problem of interpreting the effect of a given value of MSE on error rate for VSB and SQAM signaling however. Research on extensions of the present approach to error probabilities is going on now, and is expected to resolve some of these questions.

Example 1: Carrier Phase Error — Reflecting a front-end carrier phase error of θ into the channel gives $\dot{a}=0$ and $\dot{b}=-1$, so for VSB with $m_{11}(f)=1$, $B(f)=1/2$, $V(f)=0$ and $H_{V\,min}=2(1-\gamma_V-\beta_V)$, agreeing with Ref. (9). For SQAM, $B(f)=1/2$ for any staggering p, and $H_{Q\,min}=4(1-\gamma_Q\sin^2 p)$. This shows that the optimum staggering is $t_o=T/2$, a result previously found in Ref. (11). Using (21) we see that H_Q is just twice H_V, as might be expected since SQAM uses two data streams, and we add their individual mean squared errors.

Example 2: Sampling Time Error — Sampling the output θ seconds too late is easily shown to yield $\dot{a}=j2\pi f$, $\dot{b}=0$, so for both SQAM (with any staggering t_o) and VSB, $B(f)=1/2-fT$, and when $m_{11}(f)=1$, $H_{Q\,min}=(2\pi/T_Q)^2(1-\gamma_Q)^3/3$, and $H_{V\,min}=1/2(2\pi/T_V)^2(1-\gamma_V)^3/3$. The result is new for SQAM, but is well-known for VSB.(9) From (13) the effect of timing error θ on MSE is $1/2\theta^2 H_{11}$, so if random errors θ_Q/T_Q and θ_V/T_V have the same variance D, then the expected MSE values are $N_Q+(2\pi)^2(1-\gamma_Q)^3 D/3$ for SQAM and $N_V+1/2(2\pi)^2(1-\gamma_V)^3 D/3$ for VSB. Besides the understandable factor of two, VSB suffers because $\gamma_V<\gamma_Q$. The vestigial bandwidth is of no use here.

Example 3: Quadratic Delay Distortion — From (12) we see that $\dot{a}=jf(f^2+3g^2)$, $\dot{b}=g(3f^2+g^2)$. For SQAM we assume g=0 (the carrier frequency is at center of the 'delay bowl') and obtain for any t_o

$$B(f) = 1/2\left[1-fT\,\frac{3+(2fT)^2}{1/2+6(fT)^2}\right] \tag{22}$$

and $H_{Q\,min}=(4/7)(1-\gamma)^7(1/2T)^6$. Lucky(3) has found optimally tolerant pulse-shapes for quadratic delay distortion using a peak distortion criterion for baseband or DSB signaling. For the 100% excess bandwidth case and T=2 he gives for the optimum shape $B_L(f)=1/2-2f/3-f/(10.142f^2+.11624)$. In his terms (22) becomes $B_Q(f)=1/2-2f/3-f/(9f^2+.1875)$, which is a similar shape, as shown in Figure 3. It should be noted that $B_Q(f)$ does not become negative. It is clear that the optimum pulse has reduced energy at higher frequencies, since this is where the distortion is worst.

For VSB it is not so clear where the carrier frequency should be, so keeping g we obtain

$$B(f)=1/2(1-2fT/r)-(2fT/r)\frac{r^2-(2fT)^2}{r^2+3(2fT)^2},\quad V(f)=\frac{|f|}{g}\cdot\frac{3g^2+f^2}{3f^2+g^2} \tag{23}$$

where $r=1-2gT$. If $g=0$, $B(f)$ becomes the same as (22). Now from (21)

$$H_{v\ min}=(2/7)(1/2T)^6((r-\gamma)^7-(r-(1-\beta))^7) \tag{24}$$

which has its minimum when $(r-\gamma)^6=(r-(1-\beta))^7$. For a given γ and β a solution of this equation specifies the best location for the carrier frequency when f_1, the center of the delay bowl is known.

Example 4: Attenuation Slope — Here the passband channel is $e^{\theta(f-f_1)}$, so $a=e^{\theta g}\cosh(\theta f)-1$, $b=-je^{\theta g}\sinh(\theta f)$, and hence $\dot{a}=g$, $\dot{b}=-jf$. For SQAM, (14) yields for any g

$$B(f)=1/2-fT/((2fT)^2A+1-A) \tag{25}$$

where $A=\sin^2(\pi t_o/T)$. Figure 4 shows $B(f)$ for various cases, and we note that for staggerings near T/2, $B(f)$ becomes impractical. Now

$$H_{Q\ min}=4g^2+(1-\gamma)^3/3T^2+H_{extra} \tag{26}$$

where $H_{extra}=0$ if $A=0$, while for $A\neq 0$

$$H_{extra}=(1/3T^2)[(1-A)\gamma^3+3\gamma(A-1/A)+(3/A)\sqrt{(1-A)/A}\tan^{-1}(\gamma\sqrt{A/(1-A)})] \tag{27}$$

Note that $H_{extra}=0$ also for $A=1$, so either $t_o=0$ or $t_o=T/2$ is optimum staggering. Figure 5 shows H_{extra} vs A, and it is clear that it is extremely sensitive to staggering times near T/2, leading one rather to use no staggering.

For VSB we have (again with $r=1-2gT$), $B(f)=1/2-r/2fT$, $V(f)=g/|f|$, and $H_{v\ min}=(r^3-(r-(1-\beta))^3+(r-\gamma)^3)/6T^2$, which in turn has its minimum when $r=\sqrt{1-\beta-\gamma}\,(\sqrt{2(1-\beta)}-\sqrt{1-\beta-\gamma})$. The optimum $B(\cdot)$ and $V(\cdot)$ are not realizable since they have singularities. These shapes could be improved by performing a constrained minimization of H_{11}, constraining signal power on the channel, or the amplitudes of $B(\cdot)$ and $V(\cdot)$ themselves.

CONCLUSION

The problem of using available excess bandwidth to design optimally immune pulses has been attacked in a unified framework which allows one to study any two-dimensional signaling scheme. The system mean squared error for any type of channel dispersion was expanded in a Taylor series, and a suitable functional of the Hessian matrix was minimized by proper design of the Nyquist and vestigial roll-off characteristics. The techniques can be applied to a large class of channel situations. Specific examples were treated, yielding new results on optimum pulseshapes for quadratic delay distortion and attenuation slope.

REFERENCES

1. Lucky, R. W., Salz, J., and Weldon, E.S., Jr., "Principles of Data Communication," McGraw-Hill, New York, 1968.
2. Foschini, G. J., et al, "On the Selection of a Two-Dimensional Signal Constellation in the Presence of Phase Jitter and Gaussian Noise," B.S.T.J. 52, July-August 1973, pp. 927-965.
3. Lucky, R. W., "The Design of Amplitude Characteristics for Data Transmission Systems to provide immunity Against Delay Distortion," IEEE Symp. on Signal Transmission and Processing, Columbia Univ., 1965, pp. 50-56.
4. Temes, G., et al, "The Optimization of Band Limited Systems," Proc.IEEE 61, February 1973, pp. 196-234.
5. Amoroso, F. and M. Montagnana, "Distortionless Data Transmission with Minimum Peak Voltage," IEEE Trans. on Info. Theory, IT-13, July 1967, pp. 470-476.
6. Amoroso, F., "On the Efficient Use of Voice-Channel Bandwidth in Data Transmission," IEEE Trans. on Com. Tech., COM-15, October 1967, pp.669-679.
7. Tufts, D. and T. Berger, "Optimum Pulse Amplitude Modulation, Part II, Inclusion of Timing Jitter," IEEE Trans. Info. Theory, IT-13. April 1967, pp. 209-216.
8. Franks, L. E., "Further Results on Nyquist's Problem in Pulse Transmission," IEEE Trans. Com. Tech., COM-16, April 1966, pp. 337-340.
9. Hill, F. S., Jr., "Optimum PUlse Shapes for PAM Data Transmission Using VSB," IEEE Trans. on Commun., COM-23, March 1975, pp. 352-361.
10. Chiba, N., "Analysis of Optimum Pulse Waveforms in the Presence of Timing Jitter," Elec. & Comm. in Japan, 52-A, 1959, pp. 1-9.
11. Gitlin, R. D. and E. Ho, "The Performance of Staggered QAM in the Presence of Phase Jitter," IEEE Trans. on Commun., COM-23, March 1975,pp.348-352.
12. Hill, F. S., Jr., "A Unified Approach to Optimum Pulse Design in Data Transmission," in preparation.
13. Falconer, D. D. and G. J. Foschini, "Theory of Minimum MSE QAM Systems Employing Decision Feedback Equalization," BSTJ, 52, December 1973, pp. 1821-1849.

GLOSSARY OF SYMBOLS

1. Upper and lower case symbols represent Fourier transform pairs.
2. All sums and integrals extend from $-\infty$ to ∞ unless stated otherwise. $'$ = transpose, I = identity matrix, tr= trace, $\mathbf{E}$ = expectation, $*$ = complex conjugate, sgnf = 0 if f=0, and f/|f| otherwise, rep $H(f) = \sum_m H(f-m/T)$.

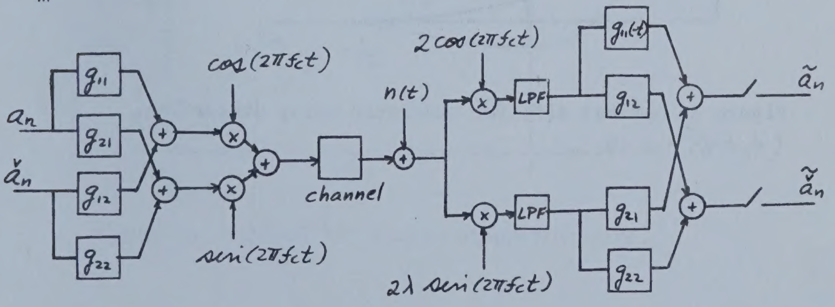


Figure 1. System Configuration.

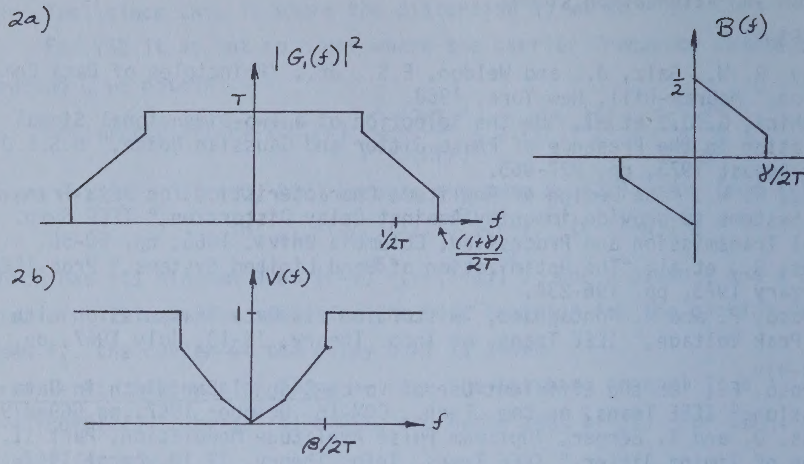

2a)
$|G_1(f)|^2$
T
f
1/2T
$\frac{(1+\gamma)}{2T}$
B(f)
$\frac{1}{2}$
f
$\gamma/2T$
2b)
V(f)
1
f
$\beta/2T$

Figure 2. Typical Roll-off and Vestigial Shapes.

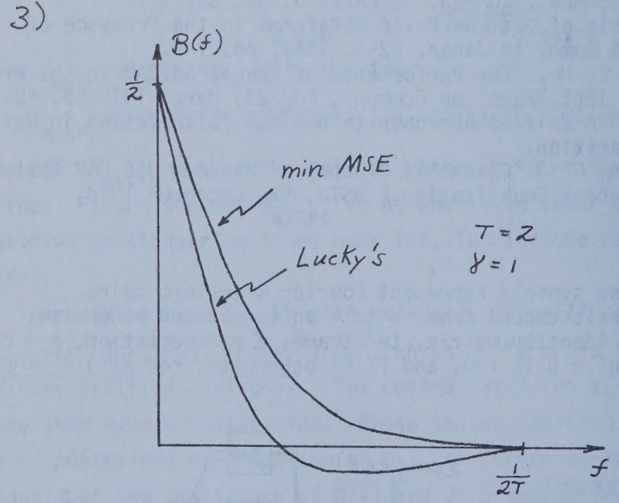


Figure 3. Optimal B(f) for quadratic delay distortion.

4)

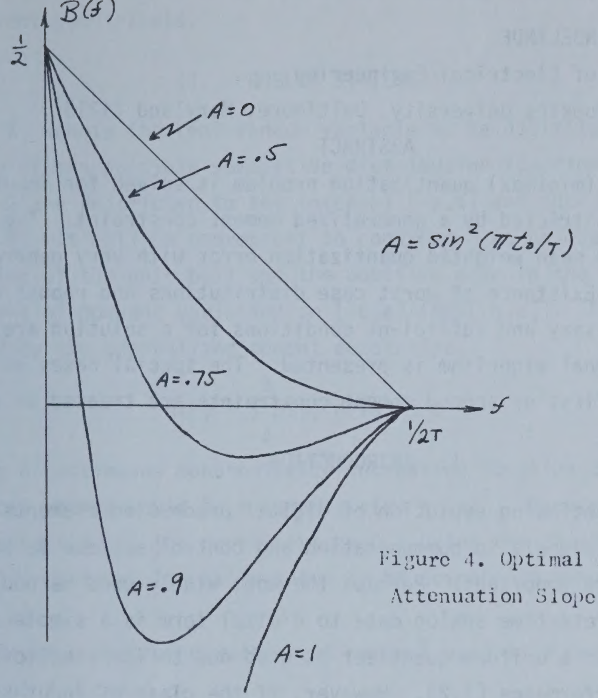


Figure 4. Optimal B(f) for Attenuation Slope Distortion

5.)

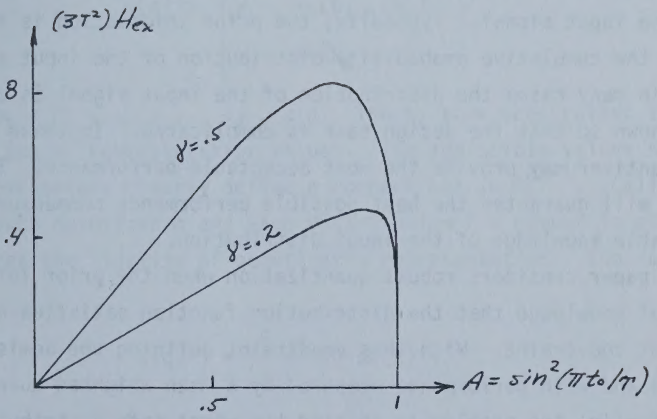


Figure 5. Residual MSE versus Staggering Time.

ROBUST QUANTIZATION WITH GENERALIZED MOMENT CONSTRAINTS

V. DAVID VANDELINDE
Department of Electrical Engineering
The Johns Hopkins University, Baltimore, Maryland 21218

ABSTRACT

The robust (minimax) quantization problem is solved for input signal distributions restricted by a generalized moment constraint. The fidelity criterion is the mean weighted quantization error with very general weighting functions. Existence of worst case distributions and robust quantizers is proved, necessary and sufficient conditions for a solution are presented and a computational algorithm is presented. The special cases of quadratic weighting with first or second moment constraints are treated as examples.

I. INTRODUCTION

With the continuing evolution of digital processing elements the task of discretizing signals in communication and control systems is becoming increasingly more important. Perhaps the most widely used method of converting discrete-time analog data to digital form is a simple amplitude quantizer. Often a uniform quantizer is used due to its simplicity and general good performance [1,2]. However, if the class of input signals can be restricted by prior information, the designer should be able to improve performance by constructing a quantizer matched to the characteristics of the input signal. Typically, the prior information is the specification of the cumulative probability distribution of the input signal. However, in many cases the distribution of the input signal is only approximately known so that the design task is complicated. In these cases a robust quantizer may provide the most acceptable performance. The robust quantizer will guarantee the best possible performance commensurate with the available knowledge of the input distribution.

This paper considers robust quantization when the prior information consists of knowledge that the distribution function satisfies a generalized moment constraint. With this constraint defining the admissible distributions and with performance measured by a mean weighted quantization error criterion, the problem is to find the worst case distribution and the best N-level quantizer in the minimax sense. Section II of the paper presents the formal statement of the problem. In Section III existence of solutions is demonstrated, necessary conditions presented and an algorithm for finding solutions is given. Section IV describes explicit solutions to the problem for quadratic performance weighting functions and first and second moment constraints.

II. PROBLEM STATEMENT

Let X denote the real random variable to be digitized and let F denote any of the possible cumulative distribution functions of X . The values of X are restricted to the interval [-a,a] and thus, so is the domain of F. It will be convenient to consider F as an element of the intersection of the unit ball and the positive cone in the normalized space of functions of bounded variation on [-a,a] (NBV[-a,a]). Furthermore, F is restricted by the generalized moment constraint,

$$G(F) = \int_{-a}^{a} \rho(s)\, dF(s) \le c \tag{1}$$

where ρ is a continuous monotonically increasing function of $|s|$. The most typical choice would be a moment, $\rho(s) = |s|^p$. Denote the set of distributions satisfying the constraint by D. Using the Banach-Alaoglu theorem [3], this set is easily shown to be compact in the weak* topology on NBV[-a,a].

A N-level quantizer q with input range [-a,a] is a device with input X and output Y defined by

$$q(x) = \begin{cases} y_1 & x\varepsilon[-a,b_1) \\ y_i & x\varepsilon[b_{i-1},b_i) \\ y_N & x\varepsilon[b_{N-1},a] \end{cases} , \tag{2}$$

with $-a \le y_1 < b_1 \le y_2 \le \dots \le y_N \le a$. The b_i have been called transition values and the y_i representation values. The admissible values of the 2N-1 quantizer parameters clearly define a compact set in R^{2N-1}. Call the set Q.

With each quantizer q and each distribution F a number is associated that measures the fidelity of quantizer's representation. The number is defined by

$$E(q,F) = \int_{-a}^{a} g^*(s)\, dF(s) , \tag{3}$$

where the integrand $g^*(s)$ is an upper semicontinuous function defined by

$$g^*(s) = \begin{cases} g(s-y_1) & s\varepsilon[-a,b_1) \\ g(s-y_i) & s\varepsilon(b_{i-1},b_i) \\ g(s-y_N) & s\varepsilon(b_N,a) \\ \max[g(b_i-y_i),\ g(b_i-y_{i+1})] & s = b_i \end{cases} , \tag{4}$$

with $g(t)$ a continuous weighting function, monotonic in $|t|$. The function $E(q,F)$, considered as a functional on D, can be shown to be weak* upper semicontinuous. Considered as a functional on Q it is continuous. However, while $E(q,F)$ is concave (linear) in F, no similar statement can be made with respect to q.

Armed with the above definitions the statement of the problem is simple:

$$\min_{q \varepsilon Q} \max_{F \varepsilon D} E(q,F). \tag{5}$$

III. SOLUTION

Before attempting to find solutions to the problem, we verify the existence of worst case distributions and optimal quantizers.

Proposition:

For an arbitrary quantizer q in Q there exists a maximizing distribution function in D. Also there exists a minimizing quantizer.

Proof:

An upper semicontinuous function achieves its supremum on a compact set [3]. Since $E(q,F)$ is weak* upper semicontinuous and D is weak* compact, the conditions are satisfied. The second part of the proposition follows, since Q is compact and the max $E(q,F)$ is lower semicontinuous.

As usual finding solutions is a bit more difficult than proving their existance. The main tool used to derive necessary and sufficient conditions is the Lagrange Duality theorem [4]. This theorem stated below in terms of the problem of the robust quantization problem.

Theorem:

Suppose that (q_0,F_0) is a solution to the robust quantization problem and $u_0 = E(q_0,F_0)$ is the optimal value of the fidelity criterion. Then the following equation holds

$$u_0 = \min_{q \varepsilon D} \min_{\lambda \geq 0} \max_{F} [E(q,F) - \lambda G(F) + \lambda c]. \tag{6}$$

Call the minimizing λ, λ_0. It is also true that

$$\lambda_0(G(F_0) - c) = 0 . \tag{7}$$

This result is extremely useful in finding solutions to the problem. Equation 6 suggests the following algorithm:

a) Maximize with respect to F. This is easily performed. Since the maximization is unconstrained, the maximizing F will be a discrete distribution placing all of its mass at the maxima of $g^*(s)-\lambda\rho(s)$.

If both g and ρ are convex (as will be the usual case), F will be concentrated on a subset of the transition values b_i (or the end points). Thus, only N+1 points need be checked to determine the largest value.

b) Minimize with respect to q and λ. This task appears to be more difficult; but, such is not the case. First, interchange the order of minimization. Next, consider the problem of minimizing with respect to the parameters of q with fixed λ. Thus, the maxima of $g^*(s) - \lambda\rho(s)$ must be made as small as possible by adjusting the parameters of q. In the case of convex g and ρ, this is easy! The parameters must be adjusted so that the values of $g^*(s) - \lambda\rho(s)$ at the transition values and the end points are equal. This will yield the quantizer parameters as a function of λ. All that remains is a one dimensional minimization with respect to λ.

The amount of effort spent in implementing the algorithm described above is quite dependent on the "niceness" of the basic functions involved in the problem statement. As the next section shows for quadratic and linear functions, the solutions are easily computed.

IV EXAMPLES

Two typical examples that illustrate the robust quantizer design are solved in this section. For both a quadratic weighting function is used, $g(s) = s^2$. In the first example $\rho(s) = |s|$ and in the second $\rho(s) = s^2$. For each case the most robust 4 and 8-level quantizers are computed. Since the problem is symmetric about the origin (for even N), there is no loss of generality by taking the domain of problem definition to be [0,1] rather than [-a,a] and reducing the number of levels by a factor of two. The computations, while best performed on a computer, are quite straightforward and require minimal effort to complete.

Example 1:

$\rho(s) = |s|$. The condition that the values of $g^*(s) - \lambda|s|$ at each of the transition values and at the end points be equal yields 2N-1 equations in 2N unknowns (one of which is λ). In this case the equations reduce to:

$$\begin{aligned} 2y_i &= b_i + b_{i-1} - \lambda, \quad i=1,\dots,N \ (b_0=0,\ b_N=1) \\ 2b_i &= y_i + y_{i+1} \quad , \quad i=1,\dots,N-1. \end{aligned} \tag{8}$$

These particular equations are easily solved, yielding for N=2

$$
\begin{aligned}
y_1 &= .25 - \lambda \\
b_1 &= .50 - \lambda \\
y_2 &= .75 - \lambda
\end{aligned}
\tag{9}
$$

For N = 4 the solutions are

$$
\begin{aligned}
y_1 &= .125 - 2\lambda \\
b_1 &= .250 - 3\lambda \\
y_2 &= .375 - 4\lambda \\
b_2 &= .500 - 4\lambda \\
y_3 &= .625 - 4\lambda \\
b_3 &= .750 - 3\lambda \\
y_4 &= .875 - 2\lambda
\end{aligned}
\tag{10}
$$

Having the quantizer values as a function of λ, all that remains is to find the correct value of λ. This is a simple one-dimensional optimization and is quickly accomplished.

Example 2:

$\rho(s) = s^2$. The equations for equality at the transition values for this example are:

$$
\begin{aligned}
2y_i &= (1-\lambda)(b_i + b_{i-1}), \quad i=1,\ldots,N \quad (b_0=0,\ b_N=1) \\
2b_i &= y_i + y_{i+1} \quad , \quad i=1,\ldots,N-1.
\end{aligned}
\tag{11}
$$

These equations are more complicated to solve analytically than the ones from the first example. However for N=2, the solution is:

$$
\begin{aligned}
y_1 &= .25\ (1-\lambda)(1-\lambda)/(1+\lambda) \\
b_1 &= .50\ (1-\lambda)/(1+\lambda) \\
y_2 &= .25\ (1-\lambda)(3+\lambda)/(1+\lambda).
\end{aligned}
\tag{12}
$$

The exact solutions for larger values of N are involved, but nonetheless the equations are linear and easily solved numerically.

Table 1 shows the dependence of the quantizer parameters on the value of the constraint bound for the symmetric four-level quantizer. For the second moment constraint the bound is expressed in terms of the standard deviation rather than the variance to facilitate comparison of the quantizer with the one determined by the first moment constraints. Of particular interest is the rapidity with which both quantizers approach the uniform quantizer. Figures 1 and 2 show the dependence of the performance on the constraint bound for symmetric four and eight-level quantizers respectively.

REFERENCES

[1] J. Max, "Quantizing for Minimum Distortion," IRE Trans. on Inform. Theory, vol. IT-6, pp. 7-12, March, 1960.

[2] J. M. Morris and V. D. VandeLinde, "Robust Quantization of Discrete-Time Signals with Independent Samples," IEEE Trans. on Communications, vol. COMM-22, pp. 1897-1902, December, 1974.

[3] R. B. Ash, Real Analysis and Probability, Academic Press, New York, 1972.

[4] D. G. Luenberger, Optimization by Vector Space Methods, Wiley, New York, 1969.

c	Linear Constraint			Quadratic Constraint		
	y_1	b_1	y_2	y_1	b_1	y_2
0.05	0.03	0.28	0.53	0.01	0.09	0.16
0.10	0.05	0.30	0.55	0.03	0.14	0.25
0.15	0.08	0.33	0.58	0.05	0.19	0.32
0.20	0.10	0.35	0.60	0.07	0.23	0.38
0.25	0.13	0.38	0.63	0.09	0.27	0.44
0.30	0.15	0.40	0.65	0.11	0.30	0.49
0.35	0.18	0.43	0.68	0.14	0.34	0.54
0.40	0.20	0.45	0.70	0.16	0.37	0.58
0.45	0.23	0.48	0.73	0.18	0.40	0.63
0.50	0.25	0.50	0.75	0.20	0.43	0.67
0.55	0.25	0.50	0.75	0.22	0.46	0.70
0.60	0.25	0.50	0.75	0.24	0.49	0.74
0.65	0.25	0.50	0.75	0.25	0.50	0.75
0.70	0.25	0.50	0.75	0.25	0.50	0.75
0.75	0.25	0.50	0.75	0.25	0.50	0.75
0.80	0.25	0.50	0.75	0.25	0.50	0.75
0.85	0.25	0.50	0.75	0.25	0.50	0.75
0.90	0.25	0.50	0.75	0.25	0.50	0.75
0.95	0.25	0.50	0.75	0.25	0.50	0.75
1.00	0.25	0.50	0.75	0.25	0.50	0.75

Table 1. Quantizer Parameter

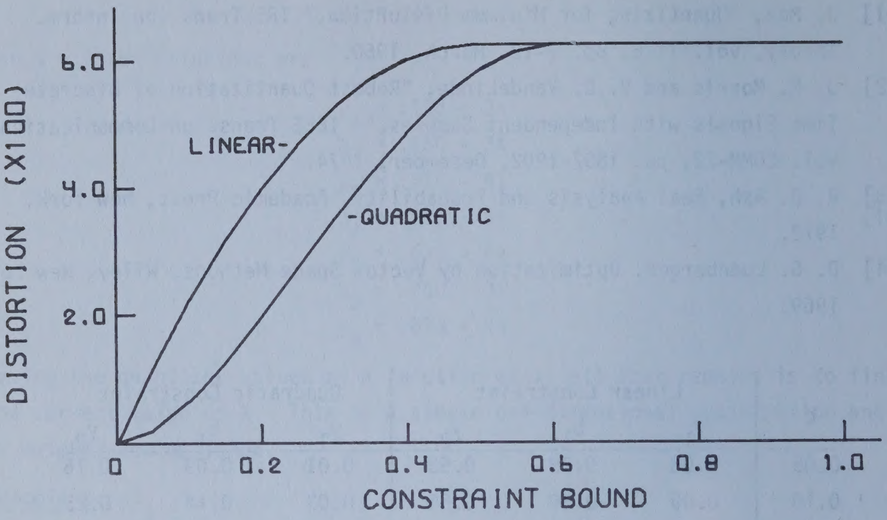


FIG. 1 FOUR-LEVEL QUANTIZER PERFORMANCE

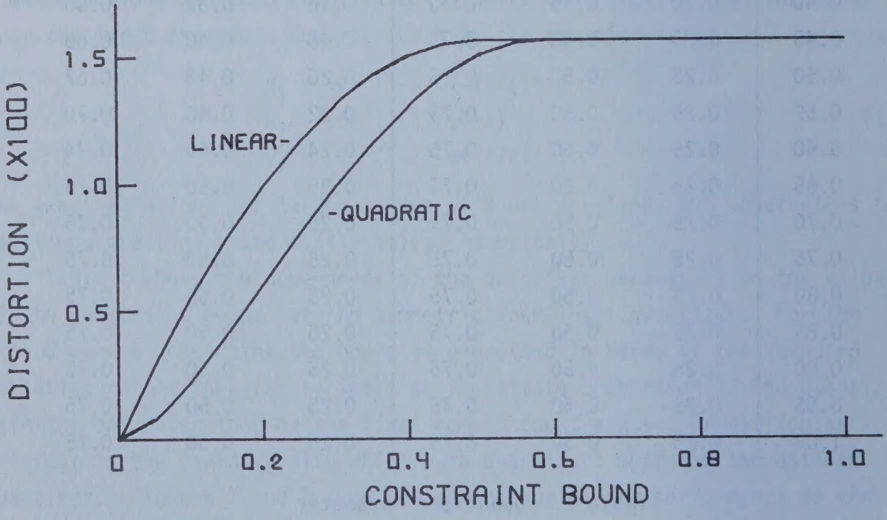


FIG. 2 EIGHT-LEVEL QUANTIZER PERFORMANCE

SELECTED TOPICS IN HIGH-SPEED DATA TRANSMISSION

RICHARD D. GITLIN
Bell Laboratories, Holmdel, New Jersey 07733

ABSTRACT

This talk surveys several aspects of in-phase and quadrature (AM/PM) high-speed (4800-9600 bits/sec) data transmission systems discussed in References [1]-[6]. The technique of passband equalization[1] will be described, along with some considerations necessary for successful digital implementation of the estimated-gradient algorithm.[2] The performance and choice of an AM/PM signal constellation in the presence of Gaussian noise and phase jitter[3],[4] will also be discussed. Various methods of data-dependent timing recovery [5],[6] will also be presented.

REFERENCES

1. R. D. Gitlin, E. Y. Ho, and J. E. Mazo, "Passband Equalization of Differentially Phase-Modulated Data Signals," BSTJ, February 1973.
2. R. D. Gitlin, J. E. Mazo, and M. G. Taylor, "On the Design of Gradient Algorithms for Digitally Implemented Adaptive Filters," IEEE Trans. on Circuit Theory, March 1973.
3. G. J. Foschini, R. D. Gitlin, and S. B. Weinstein, "On the Selection of a Two-Dimensional Signal Constellation in the Presence of Phase Jitter and Gaussian Noise," BSTJ, July 1973.
4. G. J. Foschini, R. D. Gitlin, and S. B. Weinstein, "Optimization of Two-Dimensional Signal Constellations in the Presence of Gaussian Noise," IEEE Trans. on Comm., January 1974.
5. R. D. Gitlin and J. Salz, "Timing Recovery in PAM Systems," BSTJ, May-June 1971, pp 1645-1669.
6. R. D. Gitlin and J. F. Hayes, "Timing Recovery and Scramblers in Data Transmission," BSTJ, March 1975, pp 569-593.

UPPER AND LOWER BOUNDS FOR ERROR PROBABILITY IN PULSE COMMUNICATION SYSTEMS

P.J. McLane
Department of Electrical Engineering
Queen's University
Kingston, Ontario, Canada.

ABSTRACT

A popular performance index in pulse communication systems is the probability that a transmitted symbol is in error. This performance index can be difficult to compute. When this is the case, recourse is sometimes taken to an estimate provided by upper and lower bounds. Usually these bounds are quite simply computed. Their accuracy is a function of the amount of computational effort that is practical in a given application. This paper will concentrate on bounds that differ by no more than 2 dB in system signal-to-noise ratio. Here the relative error in the bounds can be quite large but, in many applications, this estimate is quite accurate. A brief treatment of computational methods to compute the system error probability will also be given.

I INTRODUCTION

In the past two years at least two surveys have appeared [1], [2] which concern the pulse communications field. This aspect of these surveys, and, indeed of this paper, are directed toward the transmission aspects of reliably sending information symbols from a source, with a finite alphabet, to an information sink. Utilization of the information is not at issue; rather, that it be transmitted reliably and efficiently is of concern. The surveys noted above treat, among other subjects, modulation methods, receiver theory, adaptivity and the evaluation of the system error probability. Here, the last item will be treated for the special case of digital pulse-amplitude-modulation (PAM) over a channel whose filters all have frequency responses which are symmetric relative to the carrier frequency. Such filters are said to be conjugate symmetric.

For PAM transmission over voiceband channels a significant transmission impairment is intersymbol interference. This results from sending pulses at such a rate that the effective filtration that a pulse train confronts tends to smear its component pulses together in time. That is, the transmission is somewhat less than distortionless. In broadband channels, such as terrestrial microwave or satellite communications links, this type of interference is not that significant. Here carrier phase jitter and co-channel interference are the greatest limitation to reliable communication. Of course, all systems must contend with the proverbial broadband thermal noise. This paper will be concerned with all of the interferences noted above except phase jitter. To date, only brute force techniques or limited studies have appeared on the phase jitter problem.

There presently exist two distinct approaches to the problem of estimating system error probability. This mainly concerns the case of thermal noise and intersymbol interference as interferences. An early paper [4] showed that the error probability in a system where only a finite number of pulses, N, interfere, could be written as a sum of L^N values (L is the number of source symbols) of the complementary error function. When N and/or L is large, such a summation is not practical; consequently recourse is usually taken to approximations.

In these approximations the distinction to be drawn below was first referred to in [3]. One group of contributions in this area [5] - [18] dwell on upper and/or lower bounds which are simply computed: for instance they could be evaluated on a small, but scientific, pocket calculator. Here, a formula for the bound is often given.

In another group of papers [19] - [28], infinite series or computer algorithms have been developed to estimate the system error probability. Here the interest is in the numerical value of the error probability and these representations usually yield a superior numerical estimate relative to the formula-type bounds.

Both of these approaches are important. For instance, 1 dB of signal-to-noise-ratio (SNR) in a satellite communication link can be very important, whereas in voiceband systems it is relatively unimportant. Thus, in the former, an accurate estimate of error probability is needed while, in the latter estimates which differ by no more than 1 or 2 dB in SNR are quite adequate. Lucky [1] correctly notes that the methods developed for error probability estimation, or indeed the estimates themselves, have not received much use outside of error probability estimation. For instance, optimization schemes for, say the input pulse shape, with error probability as an objective function, have not come forward. Consequently aside from the odd author who compares his result with others, the general utility of the myriad of results is somewhat unknown. This paper will continue in a somewhat Chauvinistic line but an attempt will be made to put other contributions in perspective. However, time allowed only a few comparisons to be made. For these we shall use the Chebyshev pulse first used in [5].

II SYSTEM MODEL AND ERROR PROBABILITY

The system model is shown in Fig. 1. The intention is to model digital PAM transmission over a conjugate symmetric channel with carrier phase jitter, θ, co-channel interference (CCI) and thermal noise as signal impairments. It is a simple mattter to show that the input to the quantizer is

$$\hat{a}_k = a_o h_o \cos\theta + d + n \qquad (1)$$

where d is the disturbance

$$d = \sum_{i=1}^{N} a_i h_i \cos\theta + \sum_{i=1}^{M} A_i \cos\alpha_i \qquad (2)$$

with a_i one of the transmitted symbols, $h_i = h(iT)$, $h = h_T * h_c * h_r$, with * denoting convolution and A_i and α_i are respectively the amplitude and phase of the i^{th} co-channel interferer. The first component is the intersymbol interference and the second is the co-channel interference. A_i and α_i are independent but the a_i are not necessarily independent.

We shall first determine the probability of error conditioned on d and we take θ=0. The tramitted symbol levels are the 2L levels of a_k:

$$a_k = \pm 1, \pm 3, \ldots, \pm(2L-1)$$

and the quantization levels for detection are

$$0, \pm 2h_o, \pm 4h_o, \ldots, \pm(2L-2)h_o$$

When $a_o = +(2L-1)$ an error occurs when

$$d + n \leq -h_o \qquad (3)$$

If d and n are symmetric random variables this is equivalent to

$$d + n \geq h_o \qquad (4)$$

which is the condition for an error when $a_o = -(2L-1)$. For $-(2L-1)<a_o<2L-1$

an error occurs when the following event occurs:

$$E = \{d+n \geq h_o\} \cup \{d+n \leq -h_o\} \tag{5}$$

Since d and n are symmetric the two events in the union above have the same probability and thus

$$P(E) = 2P(d+n \geq h_o \tag{6}$$

Thus from (3) - (6),

$$P(e| \ a_o,d) = 2(2L-2) \ P(d+n \geq h_o) + 2P(d+n \geq h_o)$$

and as there are 2L equally likely values of a_o,

$$P(e|d) = \frac{2L-1}{L} \ P(d+n \geq h_o) \tag{7}$$

Let $\tilde{P}_e = \frac{2L-1}{L} P(e)$. Then

$$\tilde{P}(e) = E\{F_n(h_o-d)\} \tag{8}$$

where

$$F_n(x) = P(n \geq x).$$

The result in (8) is valid providing the symbols are equally likely and the random variables d and n are symmetric.

III LOWER BOUNDS

Our main lower bound is based on Jensen's inequality [29, p. 153]. Let $\theta = 0$ in (2). Further let a_i be independent. In this case $\sum_{i=1}^{N} a_i h_i$ has L^N equally likely realizations and is symmetric. One component of the CCI term $A_i \cos \alpha_i$ is symmetric and as all the terms are independent the symmetry of the sum follows from the convolution theorem for independent probability density functions. This is proved in detail in [18]. The two components of d in (2) are independent and symmetry follows by the same theorem.

Since d is a symmetric random variable

$$\tilde{P}_e = \tfrac{1}{2}[E\{F_n(h_o+|d|)\} + E\{F_n(h_o - |d|)\}]$$

and if $h_o \pm |d| \ \varepsilon \ X$, the set on which $F_n(x)$ is convex, by Jensen's inequality,

$$\tilde{P}_e \geq \tfrac{1}{2}\{F_n[h_o+E(|d|)] + F_n[h_o-E(|d|)]\}$$

Also (see [12]) for $\sigma_d^2 \triangleq E(d^2)$,

$$E(|d|) \leq \sigma_d, \qquad E(|d|) \geq \frac{\sigma_d^3}{\sqrt{E(d^4)}}$$

Thus,

$$P_e \geq \frac{1}{2} \ F_n(h_o+\sigma_d) + F_n(h_o-b) \tag{9a}$$

where

$$\sigma_d^2 = E(d^2), \qquad b = \frac{\sigma_d^3}{E(d^4)^{1/2}} \tag{9b}$$

which is a lower bound on $\tilde{P}_e$.

For the case when n is Gaussian, i.e.

$$F_n(x) = Q(\frac{x}{\sigma}) = \frac{1}{\sqrt{2\pi}} \int_{\frac{x}{\sigma}}^{\infty} \exp(\frac{-t^2}{2})dt$$

Matthews [11] has developed the bound

$$\tilde{P}_e \geq \frac{1}{2}\{Q(\frac{h_o+\sigma_d}{\sigma}) + Q[\frac{h_o-\sigma_d}{\sigma}]\} \tag{10}$$

which is subject to additional conditions given in Table 1 of his work. Since $E(d^4) \geq E(d^2)^2$, $b \leq \sigma_d$ in (9b) and when it holds Matthews bound is tighter than that in (9a). However, in many cases [12] there is little difference between the bounds and the extra work in checking the validity of (10) may not be warranted. Here (9) is valid when Q(x) is convex, i.e. when $x \geq 0$ or when $d_{max} \leq h_o$, the so-called condition for the system eye function. $h_o + d$, to be open: in other words, for n = 0, the transmission is error free.

IV UPPER BOUNDS

The Chernoff bound (30, p. 127) is often used to upper bound probabilities of the form in (7). For instance

$$P\{d+n \geq h_o\} = P\{e^{s(d+n)} \geq e^{sh_o}\}$$

for $s \geq 0$. By the Chebyshev inequality, $P(|x| \geq \varepsilon) \leq \varepsilon^{-1}E|x|$), we have

$$P(d+n \geq h_o) \leq e^{-sh_o} E[e^{sd}] \exp(s^2\sigma^2/2)$$

The last term assumes n is Gaussian. $\Phi(s)=E(e^{sd})$ is the characteristic function of d. The bound is easily worked out but it, by virtue of a bad property of the Chebyshev inequality, is weak for small ε. Lugannani decomposed $P\{d+n \geq h_o\}$ as

$$P\{d+n \geq h_o\} = P(n \geq h_o) + \int_0^{\infty} [f(h_o-s) - f(h_o+s)] P(d \geq s)ds$$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(\frac{-x^2}{2\sigma^2})$$

and then used the Chebyshev inequality on $P(d \geq s)$. The resulting expression is not complex but it is of more relative complexity than others to be presented. However, a simplification [31] of Lugannani's bound is possible and is given below

$$P_e \leq Q(\eta) + \beta \exp[-\eta^2 \frac{(1-\beta^2)}{2}] \cdot$$

$$\{2\Phi(\eta\beta) - \Phi(\bar{B}(1-\beta^2)^{1/2} + \eta\beta) - \Phi(\eta\beta-\bar{B}(1-\beta^2)^{-1/2})\}$$

where $\Phi(x) = 1-Q(x)$ and

$$\eta = \frac{h_o}{\sigma} \qquad \beta^2 = \sigma_d^2(\sigma_d^2+\sigma^2)^{-1} \qquad \bar{B} = \frac{d_{max}}{\sigma} .$$

A simple upper bound was obtained by Glave [17]. He considered - as did the work of Matthews covered in the lower bound section - the functional maximization problem of finding the probability distribution function of d such that

$$E\{F_n(x-d)\} = \max$$

subject to (i) $P(d<\infty) = 1$, (ii) $d \le d_{max}$ with probability one and (iii) $E(d^2) \le \sigma_0^2$. An interesting point is that the maximizing random variable is discrete and takes on only a finite number of values when $F_n(x)$ is an analytic function. This aids evaluation. Under mild practical conditions for instance $h_o - d_{max} \ge \sqrt{3}\,\sigma$, that is the eye is open by 3 standard deviations of the noise variance, the maximizing distributions has atoms at $\pm d_{max}$ with weight $\sigma_o^2/2\,d_{max}^2$ and an atom at 0 with one minus this weight. Usually $\sigma_o = \sigma_d$ and the bound is

$$P_e \le \left(1-\left(\frac{\sigma_d}{d_{max}}\right)^2\right) Q\left(\frac{h_o}{\sigma}\right) + \frac{1}{2}\left(\frac{\sigma_d}{d_{max}}\right)^2 \left[Q\left(\frac{h_o-d_{max}}{\sigma}\right) + Q\left(\frac{h_o+d_{max}}{\sigma}\right)\right] \qquad (12)$$

An upper bound which in some respects depends on convexity was derived by McGee [9]. In the end, a single parameter search must be conducted to evaluate the bound. In his example McGee uses the peak value of the interference d. We will derive a value analytically which for the example to be given later is superior. A little thought shows that $\ell n Q(x)$ is upper bounded by any of its tangent lines. (Also, it is lower bounded by any of its chords leading to a lower bound but this will not be pursued here). Thus

$$\ell n\, Q(x) \le \ell n\, Q(x_o) + \frac{Q'(x_o)}{Q(x_o)}\,(x-x_o)$$

or

$$Q(x) \le Q(x_o)\, \exp\left[\frac{Q'(x_o)}{Q(x)}\,(x-x_o)\right]$$

In (8) let $d = d_o$ with h_o fixed. Then

$$P_e \le Q\left(\frac{h_o-d_o}{\sigma}\right) \exp(-\lambda h_o)\,\phi(\lambda) \qquad (13a)$$

where as before $\Phi(\lambda) = E[\exp(\lambda d)]$, the characteristic function of d, and

$$\lambda = -\frac{Q'\left(\frac{h_o-d_o}{\sigma}\right)}{\sigma Q\left(\frac{h_o-d_o}{\sigma}\right)} \qquad (13b)$$

One now evaluates, or upper bounds $\Phi(\lambda)$, and finds an optimal, i.e., a minimum upper bound, or good value of d_o.

Some new results on this bound are as follows. Assuming σ is small, i.e., large SNR h_o^2/σ^2, $Q(x)$ is well represented [32, Fig. 2.35] by the first term in its asymptotic expansion,

$$Q(x) \sim \frac{\exp(-x^2/2)}{\sqrt{2\pi}\; x} \qquad (14)$$

and using this, λ in (13b) is approximately

$$\lambda \simeq \frac{h_o - d_o}{\sigma^2} \qquad (15)$$

which was also used by McGee in his example. Often, for instance, for intersymbol interference [8], the upper bound $\Phi(\lambda) \leq \exp(.5\lambda^2\sigma_d^2)$ is useful in (13). Using this in (13) as well as (14) and (15) gives

$$P_e \leq \frac{\exp\left[\dfrac{-(h_o-d_o)^2 - 2d_o(h_o-d_o) + (h_o-d_o)^2\sigma_d^2\sigma^{-2}}{2\sigma^2}\right]}{\sqrt{2\pi} \qquad \sigma^{-1}(h_o-d_o)}$$

Minimizing the exponent over d_o gives

$$d_o = \frac{\rho_1^2}{1+\rho_1^2} h_o , \quad \rho_1 = \frac{\rho_o}{\rho_d} , \quad \rho_d = \frac{h_o}{\sigma_d} , \quad \rho_o = \frac{h_o}{\sigma}$$

Then the upper bound above can be written as

$$P_e \leq \frac{\exp\left(-\dfrac{\rho_2^2(1-\rho_1^2)}{2} - \rho_2\rho_3\right)}{\sqrt{2\pi} \quad \rho_2} \qquad (16a)$$

$$\rho_o = \frac{h_o}{\sigma} \qquad \rho_d = \frac{h_o}{\sigma_d} \qquad \rho_1 = \frac{\rho_o}{\rho_d}$$

$$\rho_3 = \frac{\rho_1^2}{1+\rho_1^2} \rho_o \qquad \rho_2 = \rho_o - \rho_3 \qquad (16b)$$

IV COMPUTATIONAL METHODS

An infinite series specification for the error probability was given almost simultaneously in [19] and [20]. It is essentialy a Taylor series representation. Let $F_n(x) = Q(\frac{x}{\sigma})$ in (8). Then expanding $Q\left(\frac{h_o+d}{\sigma}\right)$ (recall that d is a symmetric random variable) about d=0 gives

$$\tilde{P}_e = Q\left(\frac{h_o}{\sigma}\right) + \sum_{n=1}^{\infty} \frac{Q^{2n}\left(\frac{h_o}{\sigma}\right) E(d^{2n})}{(2n)!}$$

where

$$Q^n(x_o) = \frac{d^n}{dx^n} Q(x)\bigg|_{x=x_o}$$

Now by (7.1.19) of [33]

$$\frac{d^{n+1}\mathrm{erf}(x)}{dx^n} = (-1)^n \frac{2}{\sqrt{\pi}} H_n(x)\, e^{-x^2}$$

where $H_n(x)$ is the n^{th} order Hermite polynomial,

$$\mathrm{erf}(x) = 1 - \mathrm{erfc}(x)$$

and

$$\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} \exp(-t^2)dt$$

$$= 2Q(x\sqrt{2})$$

Combining the results above gives for $\tilde{P}_e$

$$\tilde{P}_e = Q(\frac{h_o}{\sigma}) + \frac{\exp(-\frac{h_o}{2\cdot\sigma^2})}{\sqrt{\pi}} \sum_{n=1}^{\infty} \frac{(2\sigma^2)^{-k}}{(2k)!} H_{2k-1}(\frac{h_o}{\sqrt{2}\sigma})\, M_{2k} \tag{17a}$$

where

$$M_{2k} = E(d^{2k}) \tag{17b}$$

the $2k^{th}$ order moment of the interference d. It is easily proved that the series converges absolutely when d_{max} is bounded [19], [20] since $M_{2k} \leq d_{max}^{2k}$. For the intersymbol interference case a recurrsion formula can be written for the M_{2k}:

$$M_{2k} = - \sum_{i=1}^{k} \binom{2k-1}{2i-1} (-1)^i\, M_{2(k-i)}\, \lambda_{2i-1} \tag{18a}$$

where

$$\lambda_{2i-1} = \frac{2^{2i}(2^{2i}-1)}{2i} |B_{2i}| \sum_{\ell=1}^{N} h_\ell^{2i} \tag{18b}$$

with B_{2i} the Bernoulli members. Of course the recurrsion formula for the Hermite polynomials also aids computation:

$$H_{k+1}(x) = x\, H_k(x) - k\, H_{k-1}(x)$$

The extension of the ideas above to other sorts of interference is the theme of [28]. There are indications (see Figs. 5 and 8 of [19] that the convergence can be quite slow. However, the author can lay no claim to having experience with the method.

Prabhu [24] has introduced a variant of the computational method above and Korn [25] has followed up on the idea. Prabhu calls his bounds simple but since they are not, at least in a transparent manner, given by simple formulae, they do not meet the criterion of simplicity used here. The idea is to divide d into a sum of two components: one puts the large interferers in one component. Bounds on the joint distribution of the two components are then used to obtain a bound on the error probability. This involves the marginal distribution of, say, the component with the larger interferers. This is obtained using a minor variant on the infinite series explained above.

One relative merit of the approach is that the interference need not be finite: that is, the number of component interferers need not be finite.

Saltzberg's [8] work, which was, in the author's view, the first in error probability estimation, can also treat the infinite interference case. This work also involves an exchange of the component roles of the interference. One component peturbs the maximum pulse height h_o and the rest is incorporated into σ^2. An algorithm is presented for doing this and also some variants [3] are apparently in use. However, the overall procedure is not felt to meet our criterion of simplicity. For more work on the infinite interference case see [34], [35].

V EXAMPLE: CHEBYSHEV PULSE

As an example of the methods presented we consider d as only intersymbol interference. Here the overall, frequency normalized, channel impulse response is given by the Chebyshev pulse [5]:

$$h(t) = A_1 \cos(\frac{\omega_1|t|}{T}) - \phi_1) \exp(-\alpha_1 \frac{|t|}{T}) + A_2 \cos(\frac{\omega_2|t|}{T} - \phi_2) \exp(-\alpha_2 \frac{|t|}{T})$$

where

$$\begin{array}{llll} A_1 = .4023 & A_2 = .7163 & \omega_1 = 2.839 & \omega_2 = 1.176 \\ \phi_1 = .7553 & \phi_2 = .1602 & \alpha_1 = .4587 & \alpha_2 = 1.107 \end{array}$$

Here $h_o = h(0) = 1$ and

$$h_\ell = h(\ell-N-1), \quad \ell = 1,2,\ldots\ldots, 2N+1, \quad \ell \neq N+1$$

in (2) and we are taking the binary signalling case; i.e., $a_k=\pm1$ and L=2. The SNR in dB will be $10 \log_{10} \sigma^{-2}$

The upper and lower bounds we have computed are shown in Fig. 2. Note that the uncertainty in the error probability can be reduced by half the dB spread in SNR by saying the error probability estimate is a curve drawn equally spaced between the best upper and lower bound. Also, note that the best simple upper bound is our modification of McGee's bound. The error probability estimation here is quite adequate. However, when the sampling time is perturbed and the SNR is high none of the bounds presented are very useful (see [12] Fig. 7). In practise a rule of thumb is to try the simple bounds and see if they fit one's needs. If not, of course, the computational approaches must be used.

CONCLUSION

A rather general treatment has been presented for estimating error probabilities in pulse communication systems. Both simple upper and lower bounding formulae and computational methods have been considered. An attempt has been made to shed some light on the aims and accomplishments of both approaches. Finally, for a rather difficult pulse, all of the simple bounding methods have been compared and an adequate estimate of the error probability presented.

REFERENCES

A: General

[1] R.W. Lucky, "A Survey of the Communication Theory Literature", IEEE Inform. Theo., vol. IT-19, pp. 725-740, November, 1973.

[2] L.E. Franks, editor, "Data Communication: Fundamentals of Baseband Transmission", in Benchmark Papers in Electrical Engineering and Computer Science, Dowden, Hutchinson, and Ross, Inc., Stroudsberg, Penn., 1974.

[3] D.D. Falconer and R.D. Gitlin, "Bounds on Error-Pattern Probabilities for Digital Communications Systems," IEEE Trans. Commun. vol. COM-20, pp. 132-139, Apr. 1972.

[4] M.R. Aaron and D.W. Tufts, "Intersymbol Interference and Error Probability", IEEE Trans. Info. Theo., vol. IT-12, pp. 26-33, Jan. 1966.

B: Formula Type Upper Bounds: Intersymbol Interference

[5] R. Lugannani, "Intersymbol Interference and Probability of Error in Digital Systems", IEEE Trans. Inform. Theory, vol. IT-15, pp. 682-688, Nov. 1969.

[6] F.E. Glave, "An Upper Bound on the Probability of Error due to Intersymbol Interference for Correlated Digital Signals", IEEE Trans. Inform. Theory, vol. IT-18, pp. 356-363, May 1972.

[7] E. Hönsler, "An Upper Bound of Error Probability in Data Transmission Systems", Nachrichtentech. Z., vol. 23, pp. 625-627, 1970.

[8] B.R. Saltzberg, "Intersymbol Interference Error Bounds with Application to Ideal Bandlimited Signaling", IEEE Trans. Inform. Theory, vol. IT-14, pp. 563-568, July 1968.

[9] W.F. McGee, "A Modified Intersymbol Interference Error Bounds," IEEE Trans. Commun. (Corresp.), vol. COM-21, p. 862, July 1973.

C: Formula Type Lower Bounds: Intersymbol Interference

[10] G.E. Forney, Jr., "Lower Bounds on Error Probability in the Presence of Large Intersymbol Interference", IEEE Trans. Commun., (Corresp.), vol. COM-20, pp. 76-77, Feb., 1972.

[11] J.W. Matthews, "Sharp Error Bounds for Intersymbol Interference", IEEE Trans. Inform. Theory, vol. IT-18, pp. 356-363, May 1972.

[12] P.J. McLane, "Lower Bounds for Finite Intersymbol Interference Error Rates", IEEE Trans. on Commun., vol. COM-22, pp. 853-857, June 1974.

D: Formula Type Bounds: Co-channel Interference

[13] A.S. Rosenbaum and F.E.Glave "An Error-Probability Upper Bound for Coherent Phase-Shift Keying with Peak-Limited Interference", IEEE Trans. Commun. vol. COM-22, pp. 6-16, Jan. 1974.

[14] J.M. Aein and R.D. Turner, "Effect of Co-channel Interference on CPSK Carriers", IEEE Trans. Commun. vol. COM-21, pp. 783-790, July 1973.

[15] V.K. Prabhu, "Error Rate Considerations for Coherent Phase-Shift, Keyed Systems with Co-channel Interference", Bell Syst. Tech. Jour., vol. 48, pp. 743-767, March 1969.

[16] -----, "Error-Probability Upper Bound for Coherently Detected PSK Signals with Co-channel Interference", Election Letters, vol. 5, pp. 383-385, August, 1969.

E: Formula Type Bounds: Multiple Interferences

[17] F.E. Glave and A.S. Rosenbaum, "An Upper Bound Analysis for Coherent Phase Shift Keying with Co-channel, Adjacent Channel and Intersymbol Interference", IEEE Trans. on Commun., vol. COM-23, pp. 586-598, June 1975.

[18] P.J. McLane, "Error Rate Lower Bounds for Digital Communication with Multiple Interferences", IEEE Trans. on Commun., vol. COM-23, pp. 539-543, May 1975.

F: Computational Methods

[19] E.Y. Ho and Y.S. Yeh, "A New Approach for Evaluating the Error probability in the Presence of Intersymbol Interference and Additive Gaussin Noise", Bell Syst. Tech. J., vol. 49, pp. 2249-2266, Nov. 1970.

[20] O. Shimbo and M.I. Celebiler, "The Probability of Error Due to Intersymbol Interference and Gaussian Noise in Digital Communication Systems", IEEE Trans. Commun. Technol., vol. COM-19, pp. 113-119, Apr. 1971.
[21] E.Y. Ho and Y.S.Yeh , "Error Probability of a Multilevel Digital System with Intersymbol Interference and Gaussian Noise", Bell Syst. Tech. Jour., vol. 50, pp. 1017-1023, March 1971.
[22] Y.S. Yeh and E.Y. Ho, "Improved Intersymbol Interference Error Bounds in Digital Systems", Bell Syst. Tech. Journal, vol. 50, pp. 2585-2599, Oct. 1971.
[23] F.S. Hill, Jr., "The Computation of Error Probability for Digital Transmission", Bell Syst. Tech. Jour., vol. 50, pp. 2055-2077, July-August, 1971.
[24] V.K. Prabhu, "Some Consideration of Error Bounds in Digital Systems", Bell Syst. Tech. Jour., vol. 50, pp. 3127-3151, Dec. 1971.
[25] I. Korn, "Bounds to Probability of Error in Binary Communication Systems with Intersymbol Interference and Dependent or Independent Symbols", IEEE Trans. on Comm., vol. COM-22, pp. 251-255, Feb. 1974.
[26] I. Korn, "Probability of Error in Digital Communication Systems with Intersymbol Interference and Dependent Symbols", IEEE Trans. Inform. Theo., vol. IT-20, pp.663-668, Sept. 1974.
[27] S. Benedetto, G. de Vincentiis and A. Luneson, "Error Probability in the Presence of Intersymbol Interference and Additive Noise for Multilevel Digital Signals", IEEE Trans. on Comm., vol. COM-21, pp. 181-190, March 1973.

G: Supporting References

[28] R. Fang and O. Shimbo, "Unified Analysis of a Class of Digital Systems in Additive Noise and Interference", IEEE Trans. Commun., vol. COM-11, pp. 1075-1091, Oct. 1973.
[29] W. Feller, "An Introduction to Probability Theory and its Application", vol. 2 New York: Wiley, 1966.
[30] R.G. Gallager, "Information Theory and Reliable Communication", New York: Wiley, 1968.
[31] P.J. McLane, "Error Analysis of Euqlizer for Digital Transmission by PAM", Queen's University, Electrical Engineering, Research Report 71-2, Kingston, Ontario, Canada, July, 1971.
[32] J.M. Wozencraft and I.M. Jacobs, "Principles of Communication Engineering", New York: Wiley, 1965.
[33] M. Abramowitz and I.A. Stegun, "Handback of Mathematical Functions", Dover Publications Inc., New York, 1965.
[34] F.S. Hill, Jr., and M.A. Blanco, "Random Geometric Series and Intersymbol Interference", IEEE Trans. Infor. Theo., vol. IT-19, pp. 326-335, May 1973.
[35] S.O. Rice, "Distribution of $\Sigma a_n/n$, a_n Randomly Equal to ±1", Bell Syst. Tech. Journal, vol. 52, pp. 1097-1103, Sept. 1973.

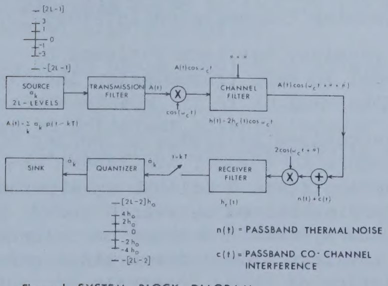

[2L-1]
3
1
0
-1
-3
-[2L-1]
SOURCE
a_k
2L-LEVELS
$A(t) = \sum_k a_k p(t-kT)$
TRANSMISSION FILTER
A(t)
X
$\cos(\omega_c t)$
$A(t)\cos\omega_c t$
CHANNEL FILTER
$h(t) = 2h_c(t)\cos\omega_c t$
$A(t)\cos(\omega_c t + \theta + \hat{\theta})$
$2\cos(\omega_c t + \hat{\theta})$
+
$n(t) + c(t)$
RECEIVER FILTER
$h_r(t)$
t=kT
$\hat{a}_k$
QUANTIZER
$\hat{a}_k$
SINK
$[2L-2]h_o$
$4h_o$
$2h_o$
0
$-2h_o$
$-4h_o$
-[2L-2]
n(t) = PASSBAND THERMAL NOISE
c(t) = PASSBAND CO-CHANNEL INTERFERENCE

Figure I: SYSTEM BLOCK DIAGRAM

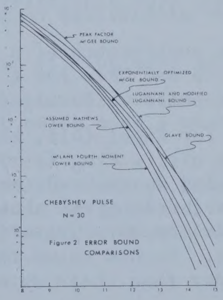


Figure 2 ERROR BOUND COMPARISONS

DIGITAL SYSTEM PERFORMANCE IN IMPULSIVE NOISE ENVIRONMENT: A SURVEY

T.H. HUYNH, M. LECOURS

Electrical Engineering Dept. Laval University, Quebec, Canada

SUMMARY

This paper is a survey of the main recent research works on digital systems performance in impulsive noise environment. Different analytical approaches are discussed and some general comments are formulated.

INTRODUCTION

Many communication channels are perturbed by non-gaussian noises: typical cases are the atmosphere (1——3), telephone lines (4——9), water (69) the earth (56). These perturbations often take the form of large amplitude pulses of any duration, and form what is called impulsive noise.

The presence of this noise in a communication channel can be due to a number of causes: lightning, switching transients, accidental hits during maintenance, microwave fades, and so on.

It is known that impulsive noise is a serious cause of errors in digital systems. However, few authors have studied the behaviour of these systems in presence of impulsive noise. This is probably due to the complexity of the analysis and the lack of information on statistical characteristics of this type of noise. The main known results are reviewed in the next section.

CLASSICAL RESULTS

The Bureau of Standards (Washington) has published a serie of measurements on the probability distribution of narrow-band impulsive atmospheric noise envelope (2, 3). From this data, Shepelavey has computed the error rate in coherent binary digital systems (36): he found that the influence of impulsive noise is less important than the influence of gaussian noise at weak S/N ratios, but becomes relatively more important at high S/N ratios. From the same data, Bello was able to compute the error probability for a binary digital system in presence of impulsive noise and fading (39).

Bodonyi has studied experimentally the error rate due to impulsive noise in function of S/N ratio for three systems: non-coherent FSK, coherent PSK and non-coherent CW (35). Results show that, of the three, non-coherent FSK has the best performance for strong impulsive noise. For small noise power, and when the pulses are widely separated in time, coherent PSK is best.

The error probability per bit has been calculated by Halsted for an autocorrelation receiver. For independant pulses occurring in time according to a Poisson distribution, he has shown that, for a given S/N ratio, the error probability decreases with increasing transmission interval.(37)

The performance of a matched filter receiver has been analysed by Lindenlaub and Chen for a simple noise model in each transmission interval only one pulse uniformly distributed in time (43).

As for Engel (41), his model is characterized by trains of pulser of the form $n(t) = K\ n_o\ (t-\tau)\ \cos\ (\omega_c t + \Psi)$ where K, τ and Ψ are random variables and ω_c the carrier frequency. The amplitude K follows an hyperbolic distribution, the phase Ψ is uniformly distributed between 0 and 2 Π. This model appears justified for narrow-band systems and noise on telephone lines. He has shown that, as long as the intervals between train pulses are longer than their duration, the statistics of these intervals are not important for his analysis. He has also shown that the performance in presence of impulsive noise can be improved by delay networks, amplitude limiters and sequential decision procedures.

Kurland and Molony (49) have measured the error probability of a 4ϕ DPSK system for different values of S/N ratio and for different pulse noise shapes, but their results were not very conclusive.

In order to compute the error probability in M-ary coherent systems, Ziemer (53, 54) has assumed a more general model including trains of pulses whose amplitude is a random variable and which occur randomly in the transmission interval according to a Poisson distribution. He has also studied the combined effect of gaussian and impulsive noise on M-ary ASK and binary PSK.

Bello and Esposito have succeeded to analyse the performance of binary PSK, DPSK and FSK, with an RF filter at the receiver front. Their results have confirmed the validity of the practice of introducing a limiter after the RF filter in order to reduce the effect of the impulsive noise (57, 58, 59).

Huynh and Lecours (67), using the same noise model as Ziemer have derived upper and lower bounds for the error probability in incoherent M-ary ASK, PSK and FSK.

Using a different noise model, in which each noise burst contains a random number of superposed waveforms, Moore and Houts have analysed the performance of binary correlation receivers. They have also derived upper bounds for the error probability which are independant of the noise and signal waveforms (62, 68).

Some researchers have also tried to study the optimum receiver against some kinds of impulsive noise. One of the first works in this direction was by Wainwright (70). He has suggested to introduce non-linear phase in the transmitted signals and then to correct them at the receiver (smearing-desmearing technique). This method could work well for noise pulses of short duration but becomes in applicable for noise bursts affective a number of consecutive symbols. Other authors (69-87) approximate the probability density by a truncated series in order to determine the optimal structure of the receiver: the find in general a so-called canonic receiver composed of a number of branches, on branch corresponding to the linear receiver and the others behaving more or less as correctors. These receivers offer an acceptable performance for near-gaussian noises.

COMMENTS

A global view of works done in the subject of impulsive noise calls for a number of comments:

- Qualitatively it is possible to conclude that impulsive noise influence of gaussian noise at low S/N ratio, and becomes relatively more important at high S/N ratios.
- One could use a limiter to attenuate the influence of impulsive noise.
- Quantitatively, it is in general not possible to compare results obtained by different authors because of the different noise models used, due in particular to the simplifying hypothesis.
- The noise models used are at both extremes: on one hand a one pulse noise model, on the other hand a near-gaussian noise (a large number of impulsive in the same transmission interval). In practice, something in between these two models would be of more interest.

- These noise models do not cover what seems to be an important case in practice: a single pulse noise affecting a number of consecutive symbols.

It would then seem important to define a more general model, for example a model of the type:

$$\sum_{k=0}^{M} h_k \quad (t, T_k)$$

when M is the number of pulses produced, $h_k(t)$ is the random waveform of the kth pulse, and T_k its instant of appearance.

This model is complex, it could for example be reduced to $h_k\ (t,T_k) = h(t,\tau_k,\ T_k)$ where $h(t,\tau,T)$ has a fixed analytical form and τ_k and τ_k are random variables representing respectively the duration and arrival time of the kth pulse.

As for the analysis of performance, it seems almost impossible to present valable results in the general case, due to the imprecision of the noise model; even for a given noise model, the analysis become quasi-unrealizable when there are many pulse in the transmission interval. It is natural then to solve the problem by Monte-Carlo simulation technique. There is then a need for a flexible program comprising a number of alternative noise models in order to study performance sensitivity to different noise models.

It is only on the base of such results one could talk realistically of optimum receivers.

As noise statistics are quite variable, depending on circonstances, adaptive structures should likely be used to obtain the best performance against impulsive noise.

REFERENCES

NOISE MEASUREMENT

1. A.D. Watt and E.L. Maxwell, "Measured statistical characteristics of VLF atmospheric radio noise," Proc. IRE, Vol. 45, pp. 55-62, Jan. 1957.

2. W.Q. Crichlow, "Noise Investigation at VLF by the National Bureau of Standard," Proc. IEEE, vol. 45, June 1957.

   W.Q. Crichlow, Q.D. Spaulding, C.J. Roubique et R.T. Disney, "Amplitude Probability Distributions for Atmospheric Radio Noise," NBS Monograph 23 November 1960.

3. W.Q. Crichlow, C.J. Roubique, A.D. Spaulding and W.M. Beery, "Determination of the amplitude-probability distribution of atmospheric radio noise from statistical moments," J.Res. Nat. Bur. Standards, vol. 64D, pp. 49-56, 1960.

4. H.L. Yudkin, "Some Results in the Measurement of Impulse Noise on Several Telephone Circuits", Proc. Nat. Elec. Conf., Chicago, Ill., 16, pp. 222-231.

5. Aikens, A.J., and Lewinski, D.A., "Evaluation of Message Circuit Noise," B.S.T.J., 39, No. 4 (July 1960), pp. 911-931.

6. R.A. Whiteman and S. Bass, "Impulse noise in data line systems," IRE Trans. Communication Systems, pp. 289-301, 1961.

7. J. Kelly, J. Mercurio, and D. Willard, "Two measuring techniques for the investigation of impulse noise and dropouts on telephone lines," AIEE Trans. (Communication and Electronics) vol. 80, pp. 585-590, January 1962.

8. Favin, D.L., "The 6A Impulse Noise Counter," Bell Laboratories Record,41 No. 3 (March 1963), pp. 100-102.

9. J.H. Fennick, "A report on some characteristics of impulse noise in telephone communication systems," IEEE Trans. Communication and Electronics, vol. 83, pp. 700-705, November 1964.

10. E.N. Skomal, "Distribution and frequency dependence of unintentionally generated man-made VHF/UHF noise in metropolitan areas," IEEE Trans. Electromagnetic Compatibility, vol. EMC-7, pp. 263-278, September 1965; "Part II: theory, ibid., pp. 420-427, December 1965.

11. "Status of information on man-made radio interference," Joint Tech. Advisory Commitee, Subcommitee 63.1.3 Unintended Radiation, July 26, 1966.

12. Kaenel, R.A., and others, "Burst Measurements in the Time Domain", Trans Audio and Electroacoustics, AU-14, No. 3 (September 1966), pp. 115-121.

13. Fennick, J.H., and Nasell, I.E.O., "The 1963 Survey of Impulse Noise on Bell System Carrier Facilities," 1966 IEEE Int. Conv. Record, Part I, Wire and Data Communication, March 1966, pp. 149-156.

14. J.E. Evans, "Preliminary Analysis of ELF Noise," M.I.T. Lincoln Laboratory Technical Note 1969-18, DDC AD 691 814, 26 March 1969.

15. A.D. Spaulding, "Amplitude and time statistics of urban man-made noise", in Proc. Int. Conf. Communications, Montreal, Canada, 1971, paper 37.

16. J.E. Evans, D.K. Willim, and J.R. Brown, "Description of the Lincoln Laboratory Wideband ELF Noise Recording Systems," M.I.T. Lincoln Laboratory, TN 1972-19, DDC AD 743 006.

STATISTICAL MODEL

17. S.O. Rice, "The Mathematical Analysis of Random Noise," B.S.T.J. vol. 23 July 1944 et vol. 24, January 1945.

18. K. Furutsu and T. Ishida, "On the theory of amplitude distribution of impulsive random noise," J. Appl. Phys., vol. 32, pp. 1206-1221, 1961.

19. P. Merts, "Model of impulsive noise for data transmission," IEEE Trans. Commun. Syst., vol. CS-9, pp. 130-137, June 1961.

20. D.A. Bowen, "Some Analytical Techniques for a Class of Non-Gaussian Processes," Research Report 63-3, Electrical Engineering Department, Queen's University, June 1963.

21. S.M. Sussman, "Analysis of the Bursts Model for Error Statistics in Telephone Circuits," IEEE Trans. Comm. Sys., 11, 1963, pp. 213-221.

22. Berger, J.M., and Mandelbrot, B., "A New Model for Error Clustering in Telephone Circuits," IBM J. Res. and Development, 7, (July 1963), pp. 222-236.

23. Kneuer, J.G., "A Simplified Physical Model for Amplitude Distribution of Impulsive Noise," IEEE, Trans. Commun. Syst., Dec. 1964, pp. 220-222.

24. O. Ibukin, "Structural aspects of atmospheric radio noise in the tropics" Proc. IEEE, Vol. 54, pp. 361-367, Mar. 1966.

25. J.H. Fennick, "Amplitude Distributions of Telephone Channel Noise and a Model for Impulse Noise," B.S.T.J., 48, No. 10 (December 1969), pp. 3243-3263.

26. D. Middleton, "Man-made noise in urban environments and transportation systems: Models and measurements," in Conf. Rec., IEEE 1972 Nat. Telecommunications Conf., Houston, Tex., paper 19C.

27. Stuck, B.W. and Kleiner, B. "A Statistical analysis of Telephone Noise", B.S.T.J., Sept. 1974, pp. 1263-1320.

28. Huynh, H.T. and Lecours, M., "Processus dePoisson filtré comme modèle du bruit impulsionnel dans les systèmes de communications digitaux", Bulletin de l'Académie Polonaise des Sciences, Série des Sciences Techniques, vol. XXII, No 2, 1974, pp. 15-20.

29. A.A. Alexander, R.M. Gryb, and D.W. Nast, "Capabilities of the Telephone Network for Data Transmission", B.S.T.J., 39, No. 3 (May 1960), pp. 431-476.

30. E.J. Hofmann, "Measured Error Distributions on the Bell A1 Facility Over Various Media," Proc. Nat. Elec. Conf., Chicago, Ill., 1960, pp. 37-44.

31. E.N. Gilbert, "Capacity of a Burst-Noise Channel," B.S.e.J., 39, No. 5 (September 1960), pp. 1253-1265.

32. A. Fontaine and R.G. Gallagher, "Error Statistics and Coding for Binary Transmission Over Telephone Circuits," Proc. IRE, 49, 1961, pp. 1059-1065.

33. E.J. Hofmann, "Error Statistics Utilizing the Code Translation Data-System Over Various Media," M.I.T. Lincoln Laboratory Report 25G-0026, 28 March 1961.

34. E.J. Hofmann, "Error Statistics Utilizing the Milgo Data System Over Various Media," M.I.T. Lincoln Laboratory Report 25G-0025, 23 January 1961

35. A.B. Bodonyi, "Effects of impulse noise on digital data transmission," IRE Trans. Communications Systems, vol. CS-9, pp. 362-366, December 1961

36. B. Shepelavey, "Non-Gaussian atmospheric noise in binary-data, phase coherent communication systems," IEE Trans. Communications Systems, vol. CS-11, pp. 280-284, September 1963.

37. L.R. Halsted, "On binary data transmission error rates due to combinations of Gaussian and impulse noise," IEEE Trans. Communications Systems vol. CS-11, pp. 428-436, December 1963.

38. A.D. Spaulding, "Determination of error rates for narrow-band communication of binary-coded messages in atmospheric radio noise," Proc. IEEE (Correspondence), vol. 52, pp. 220-221, February 1964.

39. P.A. Bello, "Error probabilities due to atmospheric noise and flat fading in HF ionospheric communications," IEEE Trans. Communication Technology, vol. COM-13, pp.266-279, September 1965.

40. A.M. Conda, "The effect of atmospheric noise on the probability of error for a NCFSK system," IEEE Trans. Communication Technology, vol. COM-13, pp. 280-285, September 1965. J.S. Engel, "Digital transmission in the presence of impulsive noise," Bell Sys. Tech. J., vol. 44, pp. 1699-1743, October 1965.

41. Fennick, J.H., "A Method for the Evaluation of Data Systems Subject to Large Noise Pulses," 1965 IEEE Int. Conv. Record, Part I, Communications 1, Wire and Data Communication, March 1965, pp. 106-110.

42. P. Mertz, "Impulse Noise and Error Performance in Data Transmission," Rand Memo RM-4526-PR, DDC AD 614 416, April 1965.

43. R.F. Lambert and R.E. Ziemer, "Square Law Detector Performance in an Impulsive Background", IEEE Trans. Com. Tech., vol. 14, no. 5, October 1966, pp.-593-594.

44. P.A. Bello, "Bounds on the error probability of FSK and PSK receivers due to non-Gaussian noise in fading channels," IEEE Trans. Information Theory, Vol. IT-12, pp. 315-326, July 1966.

45. A.D. Spaulding, "The characteristics of atmospheric radio noise and its effect on digital communication systems," presented at the IEEE Int. Commun. Conf., Philadelphia, Pa., 1966.

46. J.H. Halton and A.D. Spaulding, "Error rates in differentially coherent phase systems in non-Gaussian noise," IEEE Trans. Communication Technology, vol. COM-14, pp. 594-601, October 1966.

47. W.B. Sisco, "Bit and message error rates in atmospheric noise with and without peak limiting," Proc. NEC, vol. 23, pp. 781-786, October 1966.

48. P.A.W. Lewis and D.R. Cox, "A Statistical analysis of Telephone Circuit Error Date," IEEE Trans. Comm. Tech., 14, 1966, pp. 382-389.

49. M. Kurland and D.A. Molony, "Observations on the Effects of Pulse Noise In Digital Data Transmission Systems," IEEE Trans. Comm. Tech., 15, 1967 pp. 552-556.

50. R.E. Ziemer, "Character error probabilities for M-ary signaling in impulsive noise environment," IEEE Trans. Communication Technology, vol. COM-15, pp. 32-44, February 1967.

51. R.A. Janssen et R.F. Lambert, "Numerical Calculation of Some Response Statistics for a Linear Oscillator Under Impulsive Noise Excitation," J. Acoust, Soc. Am., vol. 41, April 1967.

52. B.A. Bowen, "The Transform Method for Non-Linear Devices with Non-Gaussian Noise," IEEE Trans. I.T., vol. 13, April 1967.

53. R.E. Ziemer, "Error Probabilities Due to Additive Combination of Gaussian of Impulsive Noise," IEEE Trans. C.T., vol. 15, June 1967.

54. R.E. Ziemer and R.A. Jassen, "Comments on the Transform Method for Non-Linear Devices with Non-Gaussian Noise," IEEE, Trans. I. T., vol. 14, July 1968,pp. 594-595.

55. Ellison, C.R., Holmstrom, J.R., and Nasell, I.E.O., "The Transmission Performance of Bell System Intertoll Trunks," B.S.T.J. 47, No. 8 (October 1968), pp. 1561-1614.

56. C. Clarisse, "Conception et Réalisation pratique d'un système de Transmission d'information à travers le sol - Application aux télémesures pétrolières," Ph. D. Thesis, l'Université de Lille, France, Juin 1969.

57. P.A. Bello and R. Esposito, "A new methot for calculating probabilities of errors due to impulsive noise," IEEE Trans. Commun. Technol. vol. COM-17, pp. 368-379, June 1969.

58. "Error probabilities due to impulsive noise in linear and hard-limited DPSK systems," IEEE Trans. Commun. Technol., vol. COM-19, pp. 14-20, February 1971.

59. M.D. Balkovic, H.W. Klancer, S.W. Klare, and W.G. McGruther, "Highspeed Voiceband Data Transmission Performance on the Switchef Telecommunications Netword," B.S.T.J., 50, No. 4 (April 1971), pp. 1349-1384.

60. J.K. Omura and P.D. Shaft, "Model performance in VLF atmospheric noise," IEEE Trans. Commun. Technol., vol. COM-19, pp. 659-668, Oct. 1971.

61. H.C. Fleming and R.M. Hutchinson, Jr., "Low-Speed Data Transmission Performance on the Switched Telecommunications Network," B.S.T._., 50, No.4 (April 1971), pp. 1385-1405.

62. J.D. Moore and R.C. Houts, "Performance of correlation receivers in the presence of impulse noise," in Conf. Rec., IEEE 1972 Nat. Telecommunications Conf., Houston, Texas, paper 15B.

63. E. Conte, E. Corti, R. Esposito, and L. Pescatori, "Error probabilities due to a mixture of impulsive and Gaussian noise in digital communication systems," Alta Freq., vol XLI, pp. 263-270, 1972.

64. A. Ephremides and L. Brandenburg, "On the Reconstruction Error of Sampled Data estimates," IEEE Trans. on Info. Thy., IT-19, 1973, pp. 365-367.

65. P.A. Bello and R. Esposito, "The Effect of impulsive noise on FSK Digital Communication," A.E.U., Band 27, pp. 25-29.

66. J.E. Ohlson, "Phase-Locked loop operation in the presence of impulsive and Gaussian Noise", IEEE Trans., Com. vol. 21, No. 9, Sept. 1973, pp. 991-996.

67. H.T. Huynh and M. Lecours, "Impulsive Noise in Non-coherent M-ary Digital Systems," IEEE Trans. Com., Vol. 23, Feb. 1975, pp. 246-252.

68. F. Moore and R.C. Houts, "Impulse Noise Performance Bounds for Binary Correlation Receivers," (Submitted for publication in IEEE Trans. Com.)

69. Middleton, "Introduction to statistical Communication Theory," Mc-Graw-Hill, 1960.

70. R.A. Wainrigh, "On the potential advantage of a Smearing-Desmearing Filter Technique in overcoming impulse-Noise Problems in Date Systems," IRE, Trans. Com. Syst., Dec., 1961.

71. D. Middleton, "Acoustic signal detection by simple correlators in the presence of non-Gaussian noise, I: Signal-to-noise ratios and canonical forms," J. Acoust. Soc. Amer., Vol. 34, pp. 1598-1609, 1962.

72. W.L. Black, "An Impulse noise Canceller," IEEE, Trans. Com. Syst., vol. C.S.J., pp. 506, Dec. 1963.

73. V.R. Algazi and M. Lerner, "Binary detection in white, non-Gaussian noise," Mass. Inst. Tech. Lincoln Lab., Mass. Inst. Tech., Cambridge Mass., M.I.T. Lincoln Lab. Rep. LL/DS-2138, 1964.

74. Y. G. Sosulin, "Optimal extraction of non-Gaussian signals in noise," Eng. Cybern. (USSR), vol. 3, pp. 89-97, 1965.

75. S.S. Rappaport and L. Kurz, "An optimal nonlinear detector for digital date transmission through non-Gaussian channels," IEEE Trans. Commun. Technol., vol. COM-14, pp. 266-274, June 1966.

76. H.M. Hall, "A new method for "impulsive" phenomena: Application of admospheric-noise communication channels," Stanford Electron, Labl, Menlo Park, Calif., Rep. SEL-66-052, 1966.

77. O.Y. Antonov, "Optimum detection of signals in non-Gaussian noise," Radio Eng. Electron. Phys. (USSR), vol. 12, pp. 541-548, 1967.

78. "Optimum detection of signals in non-Gaussian noise: Detection of a signal of unknown amplitude and phase," Radio Eng. Electron. Phys. (USSR), vol. 12, pp. 727-735, 1967.

79. R.L. Spooner, "On the detection of a known signal in a non-Gaussian noise process," J. Acoust. Soc. Amer., vol. 44, pp. 141-147, 1968.

80. V.G. Valeyev and Y. G. Sosulin; "Multichannel reception of signals in noise when the observed data have a non-Gaussian distribution," Eng. Cybern. (USSR), vol. 8, pp. 369-379, 1970.

81. "Many-channel reception of signals in noise with non-Gaussian distributions of the observed data, II," Eng. Cybern. (USSR), vol. 8, pp. 737-748, 1970.

82. D.R. Klose and L. Kurz, "A new representation theory and detection procedures for a class of non-Gaussian channels," IEEE Trans. Commun. Technol., vol. COM-17, pp. 225-234, April 1969.

83. W.J. Richter, Jr., and T.I. Smits, "Signal design and error rate of an impulse noise channel," IEEE Trans. Commun. Technol. vol. COM-19, pp. 446-458, Aug. 1971.

84. J.H. Miller and J.B. Thomas, "Detectors for discrete-time signals in non-Gaussian noise," IEEE Trans. Inform. Theory, vol. IT-18, pp. 241-250, Mar. 1972.

85. A.K. Ribin, "Classification of weak signals in non-Gaussian noise," Eng. Cybern. (USSR), vol. 10, pp. 901-909, 1972.

86. L.M. Winenberg, "Parameter Estimation for a adaptive Instrumentation of Hall's Optimum Receiver for Digital signals in Impulse noise," IEEE Trans. Com. vol. COM-22, No. 6, June 1974, pp. 798-802.

87. "Low SNR Digital Communication over certain additive non-Gaussian Channels," IEEE, Trans. Comm., vol. COM - 23, No. 3, March 1975, pp. 332-341.

A MODIFIED CONFIGURATION FOR LINEAR MULTIVARIABLE SERVOMECHANISMS*

C. A. WOLFE
Autonetics Division
Rockwell International
Anaheim, California 92803

J. S. MEDITCH
School of Engineering
University of California
Irvine, California 92664

ABSTRACT

Recent results on system type for linear multivariable servomechanisms are applied to a previously proposed type r multivariable servomechanism structure to obtain a new configuration. The latter possesses design freedom not present in the first configuration; this permits optimization of certain transient response properties of the closed-loop system without affecting previously specified system type and closed-loop pole location. An example is presented which demonstrates over a 35% improvement in integral-square tracking error relative to that for the original configuration.

I. INTRODUCTION

The notion of system type is widely known and used in classical servomechanism theory [1, 2] in both the analysis of steady-state tracking characteristics and the realization of compensation to achieve design specifications for single-input/single-output feedback control systems. Considerable progress has been made recently in extending this notion to linear multivariable (m-input/m-output) servomechanisms [3-8]. The most complete and general theory for this is given in [3-5], the earlier work in [6-8] being more specialized.

In this paper, it is shown how results from [3, 4] can be used to develop a modification of the structure of the control system in [6, 7] which possesses novel and useful design freedom not present in the original structure. The utility of this design freedom is that certain transient response properties of the closed-loop system can be optimized while neither the steady-state tracking characteristics nor the pole placement of the overall system are affected.

The paper is organized as follows. The modified control system structure is developed and discussed in Section II. In Section III, conditions are obtained for minimizing integral-square tracking error while maintaining a previously specified system type and closed-loop pole location. The results of Section III are illustrated via an example in Section IV. Finally, the conclusion is given in Section V.

II. MODIFIED SERVOMECHANISM CONFIGURATION

The approach to the linear multivariable servomechanism problem in [6, 7] uses pre-compensation (see also [4]) which integrates the error signal. This approach along with state variable feedback gives the combined state-space description

---

*Research supported in part by the U.S. Air Force Office of Scientific Research under Grant No. AFOSR 71-2116D and in part by the National Science Foundation under Grant No. GK 37735.

$$\begin{bmatrix}\dot{x}(t)\\ \dot{z}(t)\end{bmatrix} = \begin{bmatrix} A+BK & B\hat{K}\\ -\hat{B}C & \hat{A}\end{bmatrix}\begin{bmatrix}x(t)\\ z(t)\end{bmatrix} + \begin{bmatrix}0\\ \hat{B}\end{bmatrix} y_d(t)$$

$$y(t) = \begin{bmatrix} C & 0\end{bmatrix}\begin{bmatrix}x(t)\\ z(t)\end{bmatrix} \tag{1}$$

The system in Eq. (1) is a combination, in the pre-compensation configuration [4], of

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t)\\ y(t) &= Cx(t)\end{aligned} \tag{2}$$

where x(t) is an n-vector, y(t) and u(t) are m-vectors, and A, B, and C are of the appropriate dimensions, with

$$\dot{z}(t) = \hat{A}z(t) + \hat{B}e(t) \tag{3}$$

where z(t) is an mr-vector and e(t) is an m-vector, the tracking error. The control laws for u(t) and e(t) are

$$u(t) = \hat{K}z(t) + Kx(t) \tag{4}$$

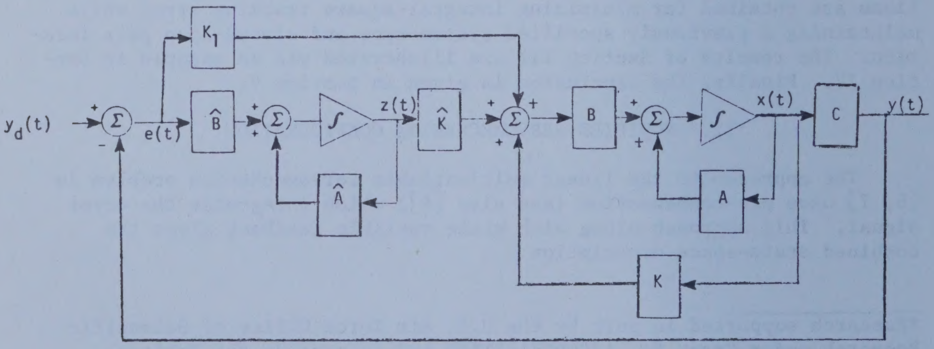
and

$$e(t) = y_d(t) - Cx(t)$$

Equation (2) represents the plant dynamics, while Eq. (3) models a compensator which performs r integrations on each of the m inputs, i.e., on e(t). Now if the output of the pre-compensator is modified to include the gain matrix $\hat{K}$ and an additional direct transmission term $K_1e(t)$, then the system of Eq. (1) becomes

$$\begin{bmatrix}\dot{x}(t)\\ \dot{z}(t)\end{bmatrix} = \begin{bmatrix} A+B(K-K_1C) & B\hat{K}\\ -\hat{B}C & \hat{A}\end{bmatrix}\begin{bmatrix}x(t)\\ z(t)\end{bmatrix} + \begin{bmatrix}BK_1\\ B\end{bmatrix} y_d(t) \tag{5}$$

which is depicted in Fig. 1.



y_d = command input (desired output)

y = output

Fig. 1. Modified control system configuration.

The advantage of this modification is that the selection of the closed-loop poles of the system does not specify the locations of all of the plant poles. The exact number of plant poles which is not specified is related to the output feedback pole placement problem [9]. The next section demonstrates how this modification can aid in the synthesis of a linear multivariable servomechanism.

III. INTEGRAL-SQUARE TRACKING ERROR REDUCTION

In this section, the direct transmission term K_1 of the pre-compensator will be used to improve tracking of step inputs during the transient period. The object will be to use the additional design freedom to reduce the transient error of the unity feedback system by a minimization of the integral-square tracking error. The unity feedback system is assumed to be a stable system which has been made at least type [1 1 ... 1] via a pre-compensator. This corresponds to the system described by Eq. (5). The gain matrix K_1 will be chosen to minimize the cost function J defined by

$$J = E\left\{\int_0^\infty [y(t) - y_d]^T [y(t) - y_d]dt\right\} = E\left[\int_0^\infty e^T(t)\ e(t)\ dt\right] \tag{6}$$

where y(t) is the m-vector of outputs, y_d is an m-vector of step inputs of random amplitude, $(\cdot)^T$ denotes the transpose, and E the expected value. Further, x(0) and y_d are taken to be zero mean, gaussian, random vectors which are independent of each other and have covariance matrices $X_o \geqslant 0$ and $Y > 0$, respectively.

For the development, it is convenient to rewrite Eq. (5) as

$$\begin{aligned} \dot{x} &= Fx + Gy_d \\ y &= Cx \end{aligned} \tag{7}$$

where the definitions of F and G are clear from Eq. (5), and F is a stable matrix which remains fixed as K_1 varies, i.e., the sum $(K - K_1C)$ remains constant.

Let $\phi(t, 0)$ represent the state transition matrix corresponding to F. Then

$$y(t) - y_d = \phi(t, 0)\ x(0) + \int_0^t C\ \phi(t, \tau)\ Gy_d\ d\tau - y_d \tag{8}$$

Since F is a stable matrix, F^{-1} exists and Eq. (8) can be expressed as

$$\begin{aligned} y(t) - y_d &= \phi(t, 0)\ x(0) + \int_0^t CF^{-1} \frac{d}{dt} \phi(t, \tau)\ Gy_d\ d\tau - y_d \\ &= \phi(t, 0)\ x(0) + CF^{-1}\ \phi(t, t)\ Gy_d - CF^{-1}\ \phi(t, 0)\ Gy_d - y_d \\ &= \phi(t, 0)\ x(0) + CF^{-1}\ Gy_d - CF^{-1}\ \phi(t, 0)\ Gy_d - y_d \end{aligned} \tag{9}$$

Further, since the system in Eq. (7) is stable and at least type [1 1 ... 1], it follows that

$$y(\infty) - y_d = CF^{-1}\ Gy_d - y_d = 0 \tag{10}$$

Then with Eq. (10) substituted into Eq. (9),

$$y(t) - y_d = \phi(t,\ 0)\ x(0) - CF^{-1}\ \phi(t,\ 0)\ Gy_d \tag{11}$$

Substituting Eq. (11) into Eq. (6) and utilizing properties of the trace operator, one has

$$\begin{aligned} J &= E\left\{\int_0^\infty [\phi(t,0)x(0)-CF^{-1}\phi(t,0)Gy_d]^T[\phi(t,0)x(0)-CF^{-1}\phi(t,0)Gy_d]dt\right\} \\ &= E\left\{tr\left[\int_0^\infty [\phi(t,0)x(0)-CF^{-1}\phi(t,0)Gy_d][\phi(t,0)x(0)-CF^{-1}\phi(t,0)Gy_d]^T dt\right]\right\} \end{aligned} \tag{12}$$

Interchanging the trace and expected value operations and taking the expectation inside the integral, Eq. (12) becomes

$$J = tr\left[\int_0^\infty CF^{-1}\ \phi(t,\ 0)\ GYG^T\phi^T(t,\ 0)F^{-1^T}\ C^T\ dt + X\right] \tag{13}$$

where

$$X \triangleq \int_0^\infty E\left[\phi(t,\ 0)\ x(0)\ x^T(0)\ \phi^T(t,\ 0)\right] dt = \int_0^\infty \phi(t,\ 0)\ X_o\ \phi^T(t,\ 0)dt$$

$$Y = E\left[y_d\ y_d^T\right]$$

and use has been made of the fact that x(0) and y_d are independent.

Using again properties of the trace operation, Eq. (13) can be written as

$$J = tr\ X\ + tr\left[GYG^T\int_0^\infty \phi^T(t,\ 0)\ F^{-1^T}\ C^T CF^{-1}\ \phi(t,\ 0)dt\right] \tag{14}$$

Since the integral in the second term on the right in Eq. (14) can be expressed as the solution of a steady-state Lyapunov equation [10], the objective function can be written finally as

$$J = tr[X] + tr[GYG^T P] \tag{15}$$

where P satisfies

$$0 = FP + PF^T + F^{-1^T}\ C^T\ CF^{-1} \tag{16}$$

As was previously noted, the system matrix F can be considered as a constant while K_1 varies. Thus only the G matrix of Eq. (15) involves K_1 directly. Using the formulas in [11], the gradient of the cost function in Eq. (16) with respect to K_1 is found to be

$$\frac{\partial J}{\partial K_1} = 0 + \frac{\partial}{\partial K_1} tr\ [GYG^T P] \tag{17}$$

If P is partitioned conformably with G, Eq. (17) can be simplified to

$$\frac{\partial}{\partial K_1} \operatorname{tr}[GYG^T P] = \frac{\partial}{\partial K_1} \operatorname{tr} \left\{ \begin{bmatrix} BK_1Y(BK_1)^T & BK_1Y\hat{B}^T \\ \hat{B}Y(BK_1)^T & \hat{B}Y\hat{B}^T \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \right\} \tag{18}$$

$$= \frac{\partial}{\partial K_1} \operatorname{tr} \left[BK_1Y(BK_1)^T P_{11} + BK_1Y\hat{B}^T P_{12}^T + \hat{B}Y(BK_1)^T P_{12} + \hat{B}Y\hat{B}^T P_{22} \right]$$

Using again the results on the gradient of the trace function [11], Eq. (18) becomes

$$\frac{\partial}{\partial K_1} \left[GYG^T P \right] = \frac{\partial}{\partial K_1} \operatorname{tr} \left[BK_1YK_1^T B^T P_{11} + 2YK_1^T B^T P_{12}\hat{B} + \hat{B}Y\hat{B}^T P_{22} \right]$$
$$= 2YK_1^T B^T P_{11} B + 2YB^T P_{12}\hat{B} \tag{19}$$

Setting Eq. (19) equal to zero gives the necessary conditions for a minimum,

$$YK_1^T B^T P_{11} B = -YB^T P_{12}\hat{B} \tag{20}$$

By assumption Y is nonsingular so that both sides may be multiplied by Y^{-1}. Also with F nonsingular and C having rank m, P_{11} is positive definite [12] and hence, with B having rank m, so is $B^T P_{11} B$. Thus, Eq. (20) can be solved for K_1 to yield

$$K_1 = -\left[B^T \; P_{12} \; B \right] \left[B^T \; P_{11} \; B \right]^{-1} \tag{21}$$

Corollary 2 of [3] guarantees that the modified servomechanism described by Eq. (4) with K_1 given in Eq. (21) is at least the same type as the servomechanism described by Eq. (1). The exact amount of improvement gained by using the modified configuration with gain K_1 will naturally depend on the specific system under consideration.

IV. EXAMPLE

To demonstrate the above results, consider the 2-input/2-output type [0 0] system given by

$$\dot{x} = Ax + Bu$$
$$= \begin{bmatrix} 1 & 3 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 1 & 1 \end{bmatrix} u$$

$$y = Cx$$
$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x$$

This system is identical to the one in Example 2 of [6]. It is rendered type [1 1] by the addition of two states. A control law of the form in Eq. (4) is calculated in [6] so that the resulting system has a fifth-order pole at -1. The resulting stable type [1 1] system is given by

$$\begin{bmatrix} \dot{x} \\ \\ \\ \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -.56 & 1.23 & .55 & .69 & -.19 \\ -3.12 & -2.54 & -.9 & 1.38 & -.38 \\ -3.12 & -3.54 & -1.9 & 2.38 & -1.38 \\ -1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & -1.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \begin{bmatrix} x \\ \\ z \end{bmatrix} + \begin{bmatrix} 0. & 0. \\ 0. & 0. \\ 0. & 0. \\ 1. & 0. \\ 0. & 1. \end{bmatrix} y_d$$

The solution to the Lyapunov equation in Eq. (16) can be obtained numerically in a number of ways [13-15]. A solution based on [15] yields

$$P = \begin{bmatrix} .473 & -.225 & .293 & .5 & -.185 \\ -.225 & .688 & -1.09 & .185 & .5 \\ .293 & -1.09 & 2.26 & -.378 & -1.63 \\ .5 & .185 & -.378 & 1.34 & 1.02 \\ -.185 & .5 & -1.63 & 1.02 & 3.93 \end{bmatrix}$$

From this and the given system equations, K_1 is calculated as

$$K_1 = \begin{bmatrix} -.834 & .501 \\ .140 & .700 \end{bmatrix}$$

With an assumed input covariance given by

$$Y = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

the cost function with the system initially in the zero state can be computed using Eq. (15) for the system before and after the modification. For the unmodified system, the cost function is 5.27. After the modification using the computed K_1, the cost function is 3.33. This is a reduction of over 35%.

The new state variable feedback matrix can be computed from the original one by

$$K = K_0 + K_1C$$

$$= \begin{bmatrix} 1.56 & 1.77 & 1.45 \\ 1.56 & 1.77 & 1.45 \end{bmatrix} + \begin{bmatrix} -.834 & .501 & 0 \\ .140 & .700 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} .726 & 2.27 & 1.45 \\ 1.70 & 2.47 & 1.45 \end{bmatrix}$$

The modified system equations differ from the original system only in the input matrix. The modified equations are then given by

$$\begin{bmatrix} \dot{x} \\ \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -.56 & 1.23 & .55 & .69 & -.19 \\ -3.12 & -2.54 & -.9 & 1.38 & -.38 \\ -3.12 & -3.54 & -1.9 & 2.38 & -1.38 \\ -1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & -1.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \begin{bmatrix} x \\ \\ z \end{bmatrix} + \begin{bmatrix} -.834 & .50 \\ 1.67 & 1.0 \\ .694 & 1.2 \\ 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} y_d$$

V. CONCLUSION

Recent results on system type for linear multivariable servomechanisms have been applied here to a previous control system configuration to obtain a modified structure which possesses useful design freedom which is absent in the previous structure. The design freedom so obtained permits minimization of integral-square tracking error without affecting system type or closed-loop pole placement. That is, the desired system type and the location of the closed-loop poles may be specified *a priori*, each independent of the other. The integral-square tracking error can then be minimized separately without disturbing satisfaction of the other two specifications. In this respect, the present results provide a "decoupled" design procedure for meeting three system specifications. For the example given in Section IV, the integral-square tracking error is improved over 35% relative to that of the original structure. The design procedure presented in this paper should be quite useful in practice.

REFERENCES

1. H. M. James, N. B. Nichols and R. S. Phillips, *Theory of Servomechanisms*, New York: McGraw-Hill, 1947.

2. S. C. Gupta and L. Hasdorff, *Fundamentals of Automatic Control*, New York: Wiley, 1970.

3. J. S. Meditch and C. A. Wolfe, "Theory of system type for linear multivariable servomechanisms," submitted to *IEEE Trans. Automat. Contr.*

4. C. A. Wolfe and J. S. Meditch, "Pre- and post-compensation to achieve system type in linear multivariable servomechanisms," submitted to *IEEE Trans. Automat. Contr.*

5. S. Hosoe and M. Ito, "On steady-state characteristics of linear multivariable systems," *Proc. 11th Annual Allerton Conf. on Ckt. and Sys. Th.*, Univ. of Ill., Urbana, Ill., pp. 477-486, Oct. 1973.

6. P. C. Young and J. C. Willems, "An approach to the linear multivariable servomechanism problem," *Int. J. Control*, vol. 15, no. 5, pp. 961-979, 1972.

7. B. Porter and A. Bradshaw, "Design of linear multivariable continuous-time tracking systems," Intl. J. Systems Sci., vol. 5, no. 12, pp. 1155-1164, 1974.

8. N. R. Sandell, Jr. and M. Athans, "On type ℓ multivariable linear systems," Automatica, vol. 9, no. 1, pp. 131-136, Jan. 1973.

9. T. E. Fortmann, "Stabilization of multivariable systems with constant-gain output feedback," Proc. 1973 Joint Auto. Contr. Conf., Ohio State Univ., Columbus, Ohio. pp. 294-301, June 1973.

10. H. Kwakernaak and R. Sivan, Linear Optimal Control Systems, New York: Wiley-Interscience, 1972.

11. M. Athans, "The matrix minimum principle," Rept. No. ESL-R-317, Electronic Sys.,Lab., Mass. Inst. Tech., Cambridge, Mass., Aug. 1967.

12. B. D. O. Anderson and J. B. Moore, Linear Optimal Control, Englewood Cliffs, N. J.: Prentice-Hall, 1971.

13. C. F. Chen and L. S. Shieh, "A note on expanding $PA + A^T P = -Q$," IEEE Trans. Automat. Contr., vol. AC-13, no. 1, pp. 122-123, Feb. 1968.

14. E. J. Davison and F. T. Man, "The numerical solution of $A'Q + QA = -C$," IEEE Trans. Automat. Contr., vol. AC-13, no. 4, pp. 448-449, Aug. 1968.

15. P. G. Smith, "Numerical solution of the matrix equation $AX + XA^T + B = 0$," IEEE Trans. Automat. Contr., vol. AC-16, no. 3, pp. 278-279, June 1971.

OPTIMAL CONTROL OF DELAY SYSTEMS

Z. BIEN and D. H. CHYUNG
Electrical Engineering, The University of Iowa, Iowa City, Iowa 52242

ABSTRACT

The problem of optimally steering the response of a time delay system to a final function is studied, and a necessary condition is obtained.

Consider a system with a single time delay

$$\dot{x}(t) = f(x(t), x(t-h), u(t), t), \quad x(t) = \phi(t), \quad t \varepsilon [t_o - h, t_o]$$

where x is an n-dimensional state vector, u is an m-dimensional control vector, $\phi(t)$, $t_o - h \leq t \leq t_o$, is a given continuous initial function and $h > 0$ is the time delay. Let $\Psi(t)$ be a given smooth final function on $[T - h, T]$, and let Ω be the control restraint set given by

$$\Omega = \{u | u \varepsilon R^m, \quad q^i(u) \leqq 0, \quad i = 1,2,\ldots,r\}$$

where q^i is a smooth scalar function for each i. The cost functional C(u) is given by

$$C(u) = \int_{t_o}^{T} f^o(x(t), x(t-h), u(t), t)\, dt$$

where f^o is a given scalar function.

The problem is to find a bounded measurable function $u(t) \varepsilon \Omega$ on $[t_o, T]$ which steers the response from the given initial function $\phi(t)$ to the given final function $\Psi(t)$ on $[T - h, T]$ for a given finite $T > 0$ while minimizing the cost functional C(u).

Let

$$\dot{x}^o(t) = f^o(x(t), x(t-h), u(t), t), \quad x^o(t_o) = 0$$

Then the problem is equivalent to the following problem; Find an optimal control u(t) on $[t_o, T]$ such that

$$\dot{x}^o(t) = f^o(x(t), x(t-h), u(t), t), \quad x^o(t_o) = 0$$

$$\dot{x}(t) = f(x(t),x(t - h),u(t),t) \qquad \text{a.e. on} \qquad [t_o,T]$$

$$x(t) = \phi(t) \qquad \text{on} \qquad [t_o - h,t_o]$$

$$x(T) = \Psi(T)$$

$$f(x(t),x(t - h),u(t),t) = \dot{\Psi}(t) \qquad \text{a.e. on} \qquad [T - h,T]$$

$$q^i(u(t)) \leq 0 \quad , \qquad 1 = 1,2,\ldots,r \quad \text{a.e. on} \qquad [t_o,T] \quad ,$$

and $x^o(T)$ is minimum.

Under certain smoothness, continuity and regularity assumptions, one can extend the result of Neustadt in [6] to obtain the following.

Let $q = (q^1,q^2,\ldots,q^r)^T$ and let f_x, f_y, and f_u be matrices defined by

$$f_x(x_o,y,u,t) = \left\{\alpha_{ij} = \frac{\partial f^i}{\partial x^j}(x,y,u,t)\Big|_{x=x_o} \quad , \qquad i,j = 1,2,\ldots,n\right\}$$

$$f_y(x,y_o,u,t) = \left\{\alpha_{ij} = \frac{\partial f^i}{\partial y^j}(x,y,u,t)\Big|_{y=y_o} \quad , \qquad i,j, = 1,2,\ldots,n\right\}$$

$$f_u(x,y,u_o,t) = \left\{\alpha_{ij} = \frac{\partial f^i}{\partial u^j}(x,y,u,t)\Big|_{u=u_o} \quad , \ i = 1,\ldots,n \ , \ j = 1,\ldots,m\right\}$$

where $f = (f^1,f^2,\ldots,f^n)^T$. Also, for $t\varepsilon[T - h,T]$, define the set $\omega(x(\cdot),t)$ in R^m to be the set of all u in Ω such that

(i) $f(x(t),x(t - h),u,t) = \dot{\Psi}(t)$,

(ii) $f_u(x(t),x(t - h)u,t)\ f_u(x(t),x(t - h),u,t)^T$ is nonsingular,

(iii) $\{f_u(x(t),x(t - h),u,t)v \mid v\varepsilon R^m,\ q^i_u(u)v < 0$ for each $i,i = 1,2,\ldots,r,$ such that $q^i(u) = 0\} = R^n$.

Maximum Principle (Necessary Condition). Let $u^*(t)$ with response $x^{o*}(t)$, $x^*(t)$ be an optimal control. Then there exist vector functions $\mu(t) = (\mu^1(t),\mu^2(t),\ldots,\mu^r(t))$, $\nu(t) = (\nu^1(t),\nu^2(t),\ldots,\nu^n(t))$ with each component in $L_\infty[T - h,T]$ and nontrivial absolutely continuous functions $\eta^o(t),\eta(t) = (\eta^1(t),\eta^2(t),\ldots,\eta^n(t))$ on $[t_o,T]$ such that

$$(\eta^o(T),\eta(T)) \neq 0$$

$$\dot{\eta}^o(t) = 0 \ , \qquad t_o \leq t \leq T \ , \qquad \eta^o(T) \leqq 0$$

$$- \dot{\eta}(t) = \eta^o[f_x^o(x^*(t),x^*(t - h),u^*(t),t) + f_y^o(x^*(t + h),x^*(t),u^*(t+h),t+h)]$$

$$+ \eta(t)f_x(x^*(t),x^*(t-h),u^*(t),t) + \eta(t+h)f_y(x^*(t+h),x^*(t),u^*(t+h),t+h)$$

$$\text{a.e. on } [t_{o,} T-2h]$$

$$- \dot{\eta}(t) = \eta^o[f_x^o(x^*(t),x^*(t-h),u^*(t),t) + f_y^o(x^*(t+h),x^*(t),u^*(t+h),t+h)]$$

$$+ \eta(t)f_x(x^*(t),x^*(t - h),u^*(t),t)$$

$$+ [\eta(t + h) + \nu(t + h)] \ f_y(x^*(t + h),x^*(t),u^*(t + h),t + h)$$

$$\text{a.e. on } [T - 2h,T - h]$$

$$- \dot{\eta}(t) = \eta^o f_x^o(x^*(t),x^*(t - h),u^*(t),t)$$

$$+ [\eta(t) + \nu(t)] \ f_x(x^*(t),x^*(t - h),u^*(t),t) \qquad \text{a.e. on } [T - h,T]$$

$$\eta^o f_u^o(x^*(t),x^*(t - h),u^*(t),t) + [\nu(t) + \eta(t)] \ f_u(x^*(t),x^*(t-h),u^*(t),t)$$

$$+ \mu(t) \ q_u(u^*(t)) = 0 \qquad \text{a.e. on } [T - h,T]$$

$$\mu^i(t) \leq 0 \ , \quad \mu^i(t)q^i(u^*(t)) = 0 \ , \quad i = 1,2,\ldots,r \ , \quad \text{a.e. on } [T - h,T]$$

and

$$\hat{\eta}(t)\hat{f}(x^*(t),x^*(t - h),u^*(t),t) = \max_{u\varepsilon\Omega} \hat{\eta}(t)\hat{f}(x^*(t),x^*(t - h),u,t)$$

$$\text{a.e. on } [t_o,T - h]$$

$$\hat{\eta}(t)\hat{f}(x^*(t),x^*(t - h),u^*(t),t) = \max_{u\varepsilon\omega(x^*(\cdot),t)} \hat{\eta}(t)\hat{f}(x^*(t),x^*(t-h),u,t)$$

$$\text{a.e. on } [T - h,T]$$

where $\hat{\eta} = (\eta^o,\eta)$ and $\hat{f} = (f^o,f)^T$.

The above maximum principle is for the case when all components of the final function are given. Similar results can be obtained when some of the components of the final function are free. The result can also be extended to cover the case when the cost functional is given by $C(u) = g(x(T)) + x^o(T)$ for some scalar function $g(x)$. Further, it can be shown that the necessary condition obtained above is also sufficient for certain linear systems with convex cost functional and convex control restraint set.

REFERENCES

[1] Banks, H. T., and M. Q. Jacobs, An Attainable Sets Approach to Optimal Control of Functional Differential Equations with Function Space Boundary Conditions, J. Diff. Eq., 13 (1973), pp. 127-49.

[2] Banks, H. T., and G. A. Kent, Control of Functional Differential Equations of Retarded and Neutral Type to Target Set in Function Space, SIAM J. Control, 10 (1972) pp. 567-93.

[3] Banks, H. T. and A. Manitus, Application of Abstract Variational Theory to Hereditary Systems — A Survey, IEEE T-AC, Vol. AC-19,(1974), pp. 524-33.

[4] Chyung, D. H., and E. B. Lee, "On Certain Extremal Problems Involving Linear Functional Differential Equation Models," in Mathematical Theory of Control, Balakrishnan and Neustadt ed., Academic Press, N. Y., 1967.

[5] Jacobs, M. Q., and T. J. Kao, An Optimum Settling Problem for Time-lag System, J. Math. Anal. Appl., 90 (1972), pp. 1-21.

[6] Makowski, K., and L. W. Neustadt, Optimal Control Problems with Mixed Control-Phase Variable Equality and Inequality Constraints, SIAM J. Control, Vol. 12, No. 2 (May 1974), pp. 184-228.

[7] Neustadt, L. W., A General Theory of Extremals, J. of Comp. and System Science, Vol. 3, No. 1 (February 1969), pp. 57-92.

[8] Neustadt, L. W. "Sufficienty Conditions and Duality Theory for Mathematical Programming Problems in Arbitrary Linear Spaces," in Nonlinear Programming, Rosen, et al., ed., Academic Press, 1970.

[9] Lee, E. B., and L. Markus, Foundations of Optimal Control Theory, John Wiley, N. Y., 1967.

ON THE IMBEDDED SOLUTION OF LINEAR SINGULAR CONTROL PROBLEM WITH DELAY

M. JAMSHIDI*
Department of Electrical Engineering

M. RAZZAGHI
Department of Mathematics and Statistics

Pahlavi University
Shiraz, Iran

ABSTRACT

A near optimum design for linear singular control problems with time-delay is presented. The originally singular time-delay system is imbedded in a class of linear systems whose first member is both non-singular and non-delay. The approximate solution is obtained by using a truncated MacLaurin's series expansion in the imbedding parameter. A numerical example illustrates the proposed design method.

INTRODUCTION

Although a number of engineering problems are singular with time-delays, very little has appeared in the literature. Soliman and Ray [1] have obtained some necessary conditions for such systems. In 1964 Kelley [2] obtained some necessary conditions for singular systems without time-delay. Moylan and Moore [3] obtained further results by transforming the singular non-delay systems into a non-singular one.

The near optimum design for non-singular problems by means of MacLaurin's expansion has received a great deal of attention in recent years [4-6]. Inoue et al. [4] have proposed a sensitivity functions method applied to systems with small time-delay. This approach involves a MacLaurin's series expansion of control function in the delay, whose coefficients are obtained from non-delay system computations only. The method presented by Jamshidi and Malek-Zavarei [5] and Chan and Perkins [6] are also sensitivity approach in which the system equation [5] or the two-point boundary value problem [6] are expanded in terms of an imbedding parameter.

In this paper, a near-optimum design for a class of singular problems with delay is presented. The proposed method introduces an imbedding parameter in the original singular time-delay system such that when the parameter is zero the system is neither singular nor delay any more. When the parameter approaches the value of one, the original problem is obtained. The method is applied to a second-order singular control system with time-delay.

---

*Presently on leave with IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.

STATEMENT OF THE PROBLEM

The class of systems considered in this paper have the form

$$\dot{x} = Ax+Bu+Cx(t-\tau) \tag{1a}$$

$$x(t) = \alpha(t) \qquad -\tau \le t \le 0 \tag{1b}$$

$$|u| \le 1 \tag{1c}$$

where $x \epsilon R_n, u \epsilon R_m$ are the state and control vectors, A, B, and C are matrices of appropriate dimensions, $\alpha(t)$ is the state's initial function and τ is the time-delay, assumed to be constant. The problem is to find an optimal control u* such that (1) is satisfied while minimizing the singular quadratic cost functional

$$J = \frac{1}{2} \int_0^T x'Qxdt \tag{2}$$

where the prime denotes vector transposition and the matrix Q is positive semi-definite. The optimization problem as given by (1) and (2) is solved using the method suggested by Moylan and Moore [3] in which the constraint (1c) on the control vector is not considered first. Once the optimal control is obtained, the solutions which violate (1c) are discarded.

In general, the solution of the above problem is difficult to obtain. In an attempt to find an approximate solution of this problem, it is imbedded in a family of non-singular systems by introducing a positive scalar parameter $\varepsilon \epsilon [0,1]$

$$\dot{x} = Ax + Bu + \varepsilon Cx(t-\tau) \tag{3}$$

$$J = \frac{1}{2} \int_0^T (x'Qx+(1-\varepsilon)u'Ru)dt \tag{4}$$

where R is a positive definite matrix. It is noted that, for $\varepsilon=0$, the original problem is changed to a linear quadratic non-singular system. For $\varepsilon \to 1$, (3) and (4) reduce to the original problem.

SUBOPTIMAL CONTROL

The necessary conditions of optimality for (3) and (4) are given below [7].

$$\dot{x} = \frac{\partial H}{\partial P} = Ax+Bu+\varepsilon Cx(t-\tau) \tag{5a}$$

$$\dot{p} = -\frac{\partial H}{\partial x} - \frac{\partial H}{\partial x(t-\tau)}\Bigg|_{t+\tau} = Qx-A'P-\varepsilon C'P(t+\tau). \quad 0 \le t \le T-\tau \tag{5b}$$

$$= -\frac{\partial H}{\partial x} = Qx-A'P \qquad T-\tau \le t \le T \tag{5c}$$

$$0 = \frac{\partial H}{\partial u} = (\varepsilon-1)Ru+B'P \tag{5d}$$

where $H=-\frac{1}{2}(x'Qx+(1-\varepsilon)u'Ru)+P'(Ax+Bu+\varepsilon Cx(t-\tau))$ is the Hamiltonian function and P is the adjoint vector. The boundary conditions are (1b) and $P(t_f)=0$. Eliminating u from (5d), and substituting in (5a), the following two-point boundary problem results

$$\dot{x} = Ax+(1-\varepsilon)^{-1}SP+\varepsilon Cx(t-\tau) \tag{6a}$$

$$\dot{P} = Qx-A'P-\varepsilon C'P(t+\tau) \qquad 0 \le t \le T-\tau \tag{6b}$$

$$= Qx-A'P \qquad T-\tau \le t \le T \tag{6c}$$

where $S = BR^{-1}B'$.
The approximate control is found by a MacLaurin's series expansion in ε,

$$u = \sum_{i=0}^{\infty} \frac{\varepsilon^i}{i!} u^i \tag{7}$$

where $u^i = \lim_{\varepsilon \to 0} \partial^i u/\partial \varepsilon^i$ for $i=0,1,2,\ldots$. These coefficients are referred to as control sensitivity functions. The "nominal" system equations are obtained by letting $\varepsilon=0$ in (6), i.e.,

$$\dot{x}^o = Ax^o+Bu^o \qquad x^o(0) = \alpha(0) \tag{8a}$$

$$\dot{P}^o = Qx^o-A'P^o \qquad P^o(T) = 0 \tag{8b}$$

with the cost function

$$J = \frac{1}{2} \int_0^T (x'^o Qx^o+u'^o Ru^o)dt \tag{8c}$$

whose solution by linear state regulator theory [8] is known

$$\dot{x}^o = (A-SK)x^o , \qquad x^o(0) = \alpha(0) \tag{9a}$$

$$P^o = -Kx^o \tag{9b}$$

$$u^o = -R^{-1}B'Kx^o \tag{9c}$$

where matrix K is the symmetric positive-definite solution of the Riccati equation

$$\dot{K} = -A'K-KA+KSK-Q, \qquad K(T) = 0 \tag{10}$$

To find the first-order sensitivity equation, differentiate (6) with respect to ε and let $\varepsilon=0$, i.e.,

$$\dot{x}^1 = Ax^1+SP^1+\gamma^o \tag{11a}$$

$$\dot{P}^1 = Qx^1-A'P^1+\delta^o \qquad 0 \le t \le T-\tau \tag{11b}$$

$$= Qx^1-A'P^1 \qquad T-\tau \le t \le T \tag{11c}$$

with boundary conditions $x^1(0)=0$, $P^1(T)=0$ and

$$\gamma^o = SP^o + C\, x^o(t-\tau)$$

$$\delta^o = -C'\, P^o(t+\tau).$$

The control sensitivity function u^1 can be obtained by differentiating (5d), i.e.,

$$u = (1-\varepsilon)^{-1}R^{-1}B'P \tag{12}$$

with respect to ε and let $\varepsilon=0$,

$$u^1 = R^{-1}B'P^1+R^{-1}B'P^o \tag{13}$$

The sensitivity function two-point boundary-value problem (11) can be similarly transformed into an uncoupled system by letting [5],

$$P^1 = -Kx^1+g_1 \tag{14}$$

where K is the solution of the Riccati equation (10) and g_1 is the first-order adjoint vector obtained by substituting (14) and its time derivation into (11b) and equating the coefficients of powers of x^1, i.e.,

$$\dot{g}_1 = -(A-SK)'g_1+K\gamma^o+\delta^o \qquad 0 \le t \le T-\tau \tag{15a}$$

$$= -(A-SK)'g_1+K\gamma^o \qquad T-\tau \le t \le T \tag{15b}$$

where $g_1(T)=0$. With this new formulation, the first-order sensitivity functions are obtained from the following equations,

$$\dot{x}^1 = (A-SK)x^1+Sg_1+\gamma^o \tag{16a}$$

$$u^1 = -R^{-1}B'Kx^1+R^{-1}B'(g_1+P^o). \tag{16b}$$

The approximate first-order control is obtained by a truncated MacLaurin's series expansion using (7) and (16b), with $\varepsilon \to 1$,

$$u \simeq u^o+u^1 = -R^{-1}B'(Kx^1-g_1)-2R^{-1}B'Kx^o \tag{17}$$

Figure 1 shows a flow chart of the proposed sub-optimal control.

EXAMPLE

Consider the following second-order singular time-delay system,

$$\dot{x} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} x + \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} x(t-0.1) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{18}$$

$$|u| \le 1, \ x(t) = 1, \ -0.1 \le t \le 0.0$$

with cost functional,

$$J = \frac{1}{2} \int_0^1 (x_1^2 + x_2^2)\ dt \tag{19}$$

Following the formulation of (3) and (4), the imbedded system becomes,

$$\dot{x} = Ax+Bu+\varepsilon Cx(t-0.1)$$

$$J = \frac{1}{2}\int_0^1 (x'Qx + (1-\varepsilon)u^2)dt \qquad (20)$$

where matrices A,B,C and Q follow directly from (18) and (19). In an attempt to obtain a sub-optimal control outlined by Figure 1, the matrix Riccati equation (10) must be solved. This is done by using the method of partitioned transition matrix outlined by Jamshidi and Razzaghi [9], i.e.,

$$K(t) = \begin{bmatrix} \frac{1}{2} - \frac{1}{2}e^{2(t-1)} & 0 \\ 0 & (1-t)/(t-2) \end{bmatrix} \qquad (21)$$

The nominal system state, costate and control can be obtained to be,

$$x^o(t) = \begin{bmatrix} e^{-t} \\ 1-t/2 \end{bmatrix}, \quad p^o(t) = \begin{bmatrix} \frac{1}{2}e^{(t-2)} - \frac{1}{2}e^{-t} \\ (1-t)/2 \end{bmatrix}$$

$$u^o(t) = (1-t)/2. \qquad (22)$$

The next step in the solution would be to solve the first-order linear adjoint equation (15) for g_1, which is done by an IBM 370/135 digital computer. Using the resulting vector, the first-order state and control vectors sensitivity functions are obtained from,

$$\dot{x}^1 = (A-SK)x^1 + Sg_1 + \gamma^o$$

$$u^1 = -R^{-1}B'Kx^1 + R^{-1}B'(g_1-P^o)$$

where R=1, and

$$\gamma^o = \begin{bmatrix} (t-2.1)/2 \\ (t-1)/2 - e^{-(t-0.1)} \end{bmatrix}$$

Figures 2 and 3 present the state and control vectors responses, respectively.

CONCLUSION

A near-optimum control for a class of singular control systems with time-delay is presented. The method chooses an imbedding parameter which forms a family of non-delay, non-singular problems. The first member of this family, called "nominal", represents a non-singular linear quadratic problem. The approximate solution is found using a truncated MacLaurin's series expansion in the imbedding parameter. The numerical example illustrated that the proposed method is simple to implement and rather fast on the computer. The extension of the method to time varying case is also possible. There are a number of studies contingent on this work which must be continued. One is the convergence of the imbedded family of non-singular problems. The other is the necessary or sufficient condition for the existence of singular subarcs, perhaps as a function of ε. Another study is to make a comparison between the approximate solution and the exact one.

REFERENCES

1. M.A. Soliman and W.H. Ray, "On the Optimal Control of Systems Having Pure Time Delays and Singular Arcs", Int. J. Control, Vol. 16, no. 5, pp. 963-976, 1972.

2. H.J. Kelley, "A Second Variation Test for Singular Extremals", AIAA. J., Vol. 2, pp. 1380-1382, 1964.

3. P.J. Moylan and J.B. Moore, "Generalization of Singular Optimal Control Theory", Automatica, Vol. 7, pp. 591-598, 1971.

4. K.I. Inoue, H. Akashi, K. Ogino and Y. Sawaragi, "Sensitivity Approaches to Optimization of Linear System with Time Delay", Automatica, Vol. 7, pp. 671-679, 1971.

5. M. Jamshidi and M. Malek-Zavarei, "Suboptimal Design of Linear Control Systems with Time Delay", Proc. IEE, Vol. 119, no. 12, pp. 1743-1746, 1972.

6. H.C. Chan and W.R. Perkins, "Optimization of Time Delay Systems Using Parameter Imbedding", Automatica, Vol. 9, pp. 257-261, 1973.

7. G.L. Kharatishvili, "The Maximum Principle in the Theory of Optimal Processes with Time-Lags", Dolk. Akad. Nauk USSR, Vol. 136, pp. 39-43, 1961.

8. R.E. Kalman, "Contribution to the Theory of Optimal Control", Bol. Soc. Mat. Mexicana, pp. 102-119, 1960.

9. M. Jamshidi and M. Razzaghi, "Optimization of Linear Systems with Input Time-Delay", to appear in Kybernetika, Prague.

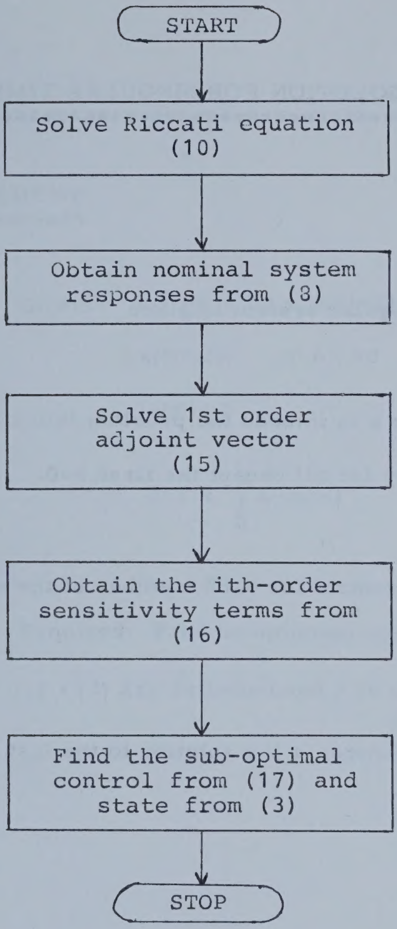

START
Solve Riccati equation (10)
Obtain nominal system responses from (8)
Solve 1st order adjoint vector (15)
Obtain the ith-order sensitivity terms from (16)
Find the sub-optimal control from (17) and state from (3)
STOP

Figure 1. A flowchart for the proposed suboptimal design.

AN IMBEDDED SOLUTION FOR SINGULAR TIME-DELAY SYSTEMS

1. Linear time-delay singular system is given

2. Introduce a parameter k to imbedd the problem into a class of linear, non-singular but time-delay for all except the first, $k=0$.

3. Find a near-optimum control for each member of the class as expanded solution of $k=0$ case.

4. The desired optimal control is the solution to the last problem, i. e. $k=1$.

PROBLEM STATEMENT

****

Given: $$\dot{x} \equiv Ax + Bu + Cx(t-T) \qquad (1)$$

$$x(t)=a(t) \qquad -T \le t \le 0 \qquad (2)$$

$$|u(t)| \le 1 \qquad (3)$$

$$J=1/2 \int_0^{t_f} x'Qx \, dt$$

Required: Find an optimum control $u^*(t)$ such that (1) - (3) are satisfied and J is also minimized

IMBEDDED CLASS
****

$$\dot{x}=Ax + Bu + kCx(t-T)$$

$$x(t)=a(t) \qquad -T \le t \le 0$$

$$J=1/2 \int_0^{t_f} (x'Qx + (1-k)u'Ru)dt$$

NOTE:

For k=0

$$\dot{x}=Ax + Bu$$

$$J=1/2 \int_0^{t_f} (x'Qx + u'Ru)dt \qquad \text{.... NOMINAL}$$

For k=1

$$\dot{x}=Ax + Bu + Cx(t-T)$$

$$J=1/2 \int_0^{t_f} x'Qx\, dt \qquad \text{............. ORIGINAL}$$

SUB-OPTIMAL CONTROL

Assuming no bounds on u(t) , the necessary optimality conditions:

$$\dot{x}=H_p=Ax + Bu + kCx(t-T)$$

$$\dot{p}=-H_x-H_{x(t-T)}\Big]_{t=t+T} = Qx - A'p - kC'p(t+T) \;\ldots\ldots\quad 0\leq t\leq t_f-T$$

$$=-H_x = Qx - A'p \qquad\qquad t_f-T\leq t\leq t_f$$

$$0=H_u = (k-1)Ru + B'p$$

Eliminating u,

$$\dot{x}=Ax + (1-k)^{-1}Sp + kCx(t-T)$$

$$\dot{p}=Qx - A'p - kC'p(t+T) \quad\ldots\ldots\ldots\quad 0 \leq t \leq t_f-T$$

$$=Qx - A'p \quad\ldots\ldots\ldots\quad t_f-T\leq t\leq t_f$$

where $S=BR^{-1}B'$

Assuming all derivatives exist, expand u(t, k),

$$u(t, k) = \sum_{0}^{\infty} (k^{i}/i!) \, . \, u^{i}$$

$$\text{where} \quad \overset{i}{u} = \lim_{k\to 0} \overset{i}{\partial} u/\partial k^{i}$$

NOMINAL System solution,

$$\overset{c}{u} = -R^{-1} B'Kx^{c}$$

$$\dot{\overset{c}{x}} = (A - SK)x^{c}$$

$$\overset{c}{p} = -Kx^{c}$$

where K(t) is the solution of the Riccati equation,

$$\dot{K} = -A'K - KA + KSK - Q \;\; , \;\; K(t_f) = 0.$$

SERIES TERMS

****

$$\dot{x}^1 = Ax^1 + Sp^1 + v^0$$

$$\dot{p}^1 = Qx^1 - A'p^1 + w^0$$

$$= Qx^1 - A'p^1$$

where

$$v^0 = Sp^o + Cx^o(t-T)$$

$$w^0 = -C'p^o(t+T)$$

let $p^1 = -Kx^1 + g_1$

$$\dot{x}^1 = (A-SK)x^1 + Sg_1 + v^0$$

$$u^1 = -R^{-1}B'(Kx^1 + g_1) + R^{-1}B'p^o$$

x^o and p^o correspond to the soution of the NOMINAL system

First-order near-optimum control is

$$u^*(t) = -R^{-1}B'(Kx^1 + g_1) \quad -2R^{-1}B'Kx^o$$

EXAMPLE
****

$$\dot{x}_1 = -x_1 - x_2(t-.10)$$

$$\dot{x}_2 = -x_2 - x_1(t-.10) + u$$

$|u| \leq 1$, $x(t)=1$, $-0.1 \leq t \leq 0.0$

Min: $J = 1/2 \int_0^1 (x_1^2 + x_2^2)\, dt$

the resulting Riccati matrix is obtained analytically:

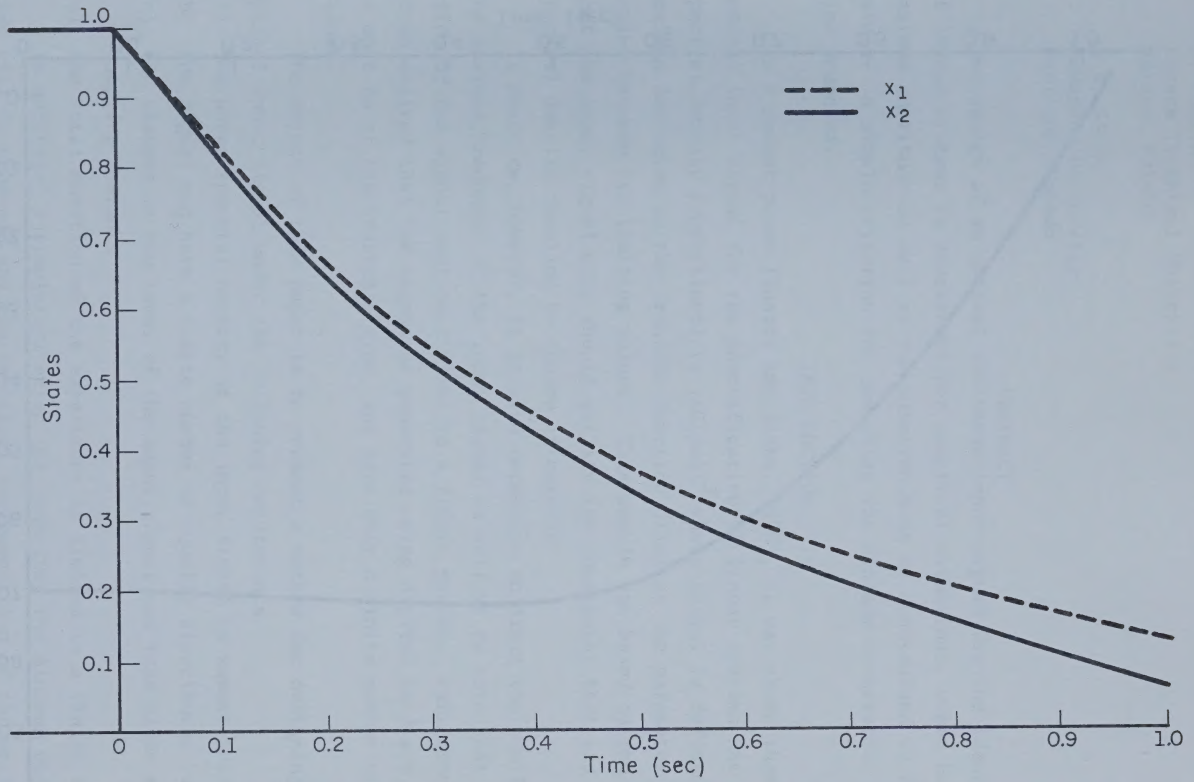
$$K(t) = \begin{bmatrix} 1/2 - 1/2\, e^{2(t-1)} & 0 \\ 0 & (1-t)/(t-2) \end{bmatrix}$$

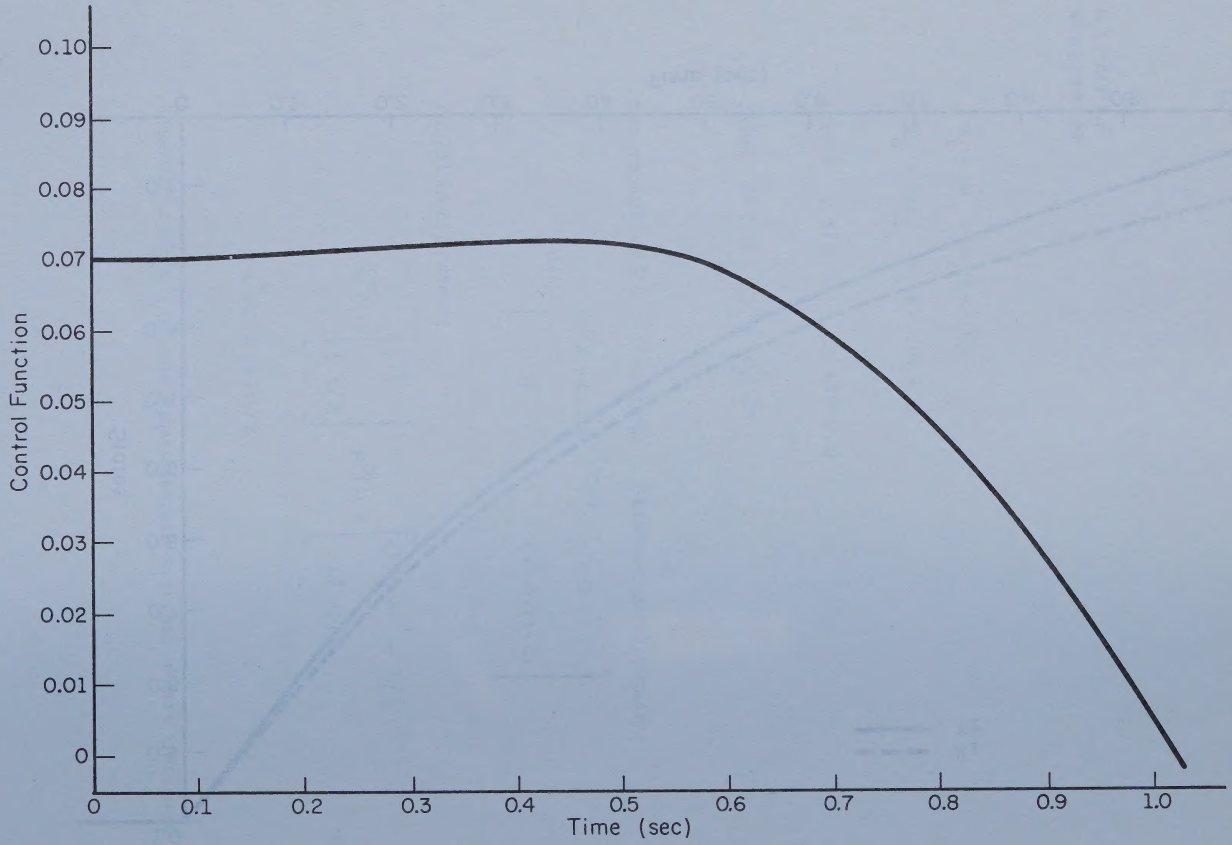
NOMINAL System:

$$x^o(t) = \begin{bmatrix} e^{-t} \\ 1-t/2 \end{bmatrix} \qquad p^o(t) = \begin{bmatrix} -1/2\, e^{-t} + 1/2\, e^{t-2} \\ (1-t)/2 \end{bmatrix}$$

$$u^o(t) = (1-t)/2$$

Similarly x^1, p^1, u^1 and approximate x, p and u^* are obtained


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
States
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Time (sec)
x_1
x_2


Control Function
0.10
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Time (sec)

OPTIMAL STAIRCASE INPUT SIGNAL FOR SYSTEM IDENTIFICATION

B. KUSZTA
Warsaw Technical University
Warsaw, Poland

N.K. SINHA
McMaster University
Hamilton, Canada

ABSTRACT

The design of an optimal staircase input signal for the identification of linear systems is considered for practical situations, where both the maximum amplitude as well as the duration must be constrained to finite values. A simple criterion for computing the minimum observation time is also proposed.

INTRODUCTION

In a recent paper [Kuszta and Sinha, 1975], it was shown that the optimal input signal for the identification of linear systems has its power spectral density proportional to $|\Delta G(j\omega)|^2$ where $\Delta G(j\omega)$ is defined as the maximum deviation in the transfer function $G(j\omega)$ as the parameter vector varies between its limiting values. This result was based on the assumption that the input signal $u(t)$ should satisfy the constraint that its power spectral density function be square integrable.

In practice, however, it is also necessary to limit the amplitude and the maximum frequency of the input signal as well as to note that the duration of the signal must be limited to a finite period. Furthermore, it is often desired that the signal be generated using digital hardware, so that it must be of the staircase type, and have only a finite number of possible values.

The object of this paper is to present a method for designing the optimal input signal under the following constraints:

(a) the power spectral density of the input signal is square integrable
(b) the signal can have a finite number of equally distributed values.
(c) the changes in the level of the input signal can take place at sampling instants
(d) the total duration of the experiment is limited to a finite time T.

In addition, a simple criterion for computing the minimum observation time will be proposed and the practical implementation of the entire scheme will be discussed.

Optimal discrete-time input for an infinite observation time

If the constraints (b), (c) and (d) had not been given, it would have been conceptually quite straightforward to generate the optimal input by applying white noise to a linear filter of transfer function $\Delta G(j\omega)$. Due to the other restrictions however, the input signal must be generated by applying a discrete-time (staircase) sequence to a digital filter, or alternatively, it may be generated digitally.

Thus, our problem is to determine the staircase signal of fixed sampling period t_s, such that the shape of its power spectral density will be as close as possible to $|\Delta G(j\omega)|^2$, while allowing the signal to have only a finite number of possible levels. Unfortunately, as far as the authors are aware, very little has been published on the generation of such signals for arbitrary power spectra. It is well-known, however, [Nikiforuk and Gupta, 1969] that white noise may be approximated by an m-sequence, whereas, coloured noise may be generated by using a weighted m-sequence [Kashiwagi 1974].

The general case of generating a staircase signal with an arbitrary power spectral density can be solved by decomposing it into, spectra of two basic types, (i) those which have a single peak at $\omega = 0$, and then decrease monotonically, and (ii) those which have a single peak at $\omega = \omega_1$ (considering positive frequencies only). Signals with spectra of the first type can be easily generated by using a weighted m-sequence as suggested by Kashiwagi [1974] with some modification, to the generation of signals with spectra of the second type.

A schematic diagram of the weighted m-sequence generator is shown in Fig. 1. Let m_i, m_{i+1}, ..., m_{i+N-1} be the values of the stages of the shift register at time $t = it_s$ (where i is an integer) and x_i^o be the value of the weighted m-sequence at time $t = it_s$, where $N = 2^n-1$ and n is the number of stages in the shift register. Then, the signal x_i^o may be expressed as follows

$$\underline{x}_i^o = M_i \underline{a} \tag{1}$$

where

$$\underline{x}_i^o \triangleq \begin{bmatrix} x_i^o \\ x_{i+1}^o \\ \cdot \\ \cdot \\ \cdot \\ x_{i+N-1}^o \end{bmatrix} \tag{2}$$

$$\underline{a} \triangleq \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_n \end{bmatrix} \tag{3}$$

and

$$M_i \triangleq \begin{bmatrix} m_i & m_{i+1} & \cdots & m_{i+n-1} \\ m_{i+1} & m_{i+2} & \cdots & m_{i+n} \\ \vdots & & & \vdots \\ m_{i+N-1} & m_{i+N} & \cdots & m_{i+N+n-2} \end{bmatrix} \tag{4}$$

The mean value of signal x_i^o is given by

$$\bar{x}^o = -\frac{1}{N} \sum_{i=1}^{n} a_i \tag{5}$$

and the autocorrelation function of the zero-mean weighted m-sequence is as follows

$$R_{xx}(kt_s) = \begin{cases} \left(\frac{N+1}{N}\right) \sum_{i=1}^{n-k} a_i \, a_{i+k} - \frac{N+1}{N}\left(\sum_{i=1}^{n} a_i\right)^2 & \text{for } k < n \\ -\frac{N+1}{N}\left(\sum_{i=1}^{n} a_i\right)^2 & \text{for } k \geq n \end{cases} \tag{6}$$

For the case when $a_i = r^{n-i}$, where r is an integer, the autocorrelation function may be approximated by

$$\phi_{xx}(kt_s) = r^{-k}, \quad k = 0,1,2,\ldots \tag{7}$$

and the envelope of the power spectral density may be approximated by

$$\Phi_{xx}(\omega) = \frac{2\,\frac{\log r}{t_s}}{\left(\frac{\log r}{t_s}\right)^2 + \omega^2} \tag{8}$$

For $r = 2$, x_i takes the value of every even number between 2^n-2 and $-(2^n-2)$, thereby satisfying condition (b).

Signals with spectra of the second type may be generated by using pseudo-ternary codes [Van Gerwen, 1965]. These may be obtained by using the following choice for the vector $\underline{a}$ in the weighted m-sequence,

$$\begin{aligned} a_1 &= 1 \\ a_\ell &= -1 \\ a_k &= 0, \quad k = 2,3,\ldots n, \; k \neq \ell \end{aligned} \tag{9}$$

The envelope of the power spectral density of the resulting sequence is given by

$$|\Phi_{xx}(\omega)|^2 = |1 - e^{-j\ell\omega t_s}|^2 = 2(1-\cos\ell\omega t_s) \tag{10}$$

A sketch of $|\Phi(\omega)|$ against ω is shown in Figure 2. To make the peak of $|\Phi(\omega)|$ coincide with ω_1, we must make

$$\omega_1 = \frac{1}{2\ell t_s} \tag{11}$$

Effect of finite period of observation

The derivations of the previous section apply to the case when the period of observation is sufficiently large, i.e., when $N\to\infty$. For a finite period of observation, the resulting truncated m-sequence does not give desirable power spectrum. The problem of selection of pulse sequences which will power spectre closely approximating white noise has been studied in detail in Communication Theory [Barker 1953, Huffman 1962], and Turyn [1968] has presented tables of sequences with small auto-correlation except at the origin. Appropriate sequences for system identification have been determined experimentally by Burger [1967] but it can be easily shown that the same result would have been obtained using Barker codes. Moreover, Turyn and others have shown that there are no other codes with this property.

To obtain power spectra other than white noise from truncated sequences, one can use the same form of weighting with the limited binary sequence as proposed for m-sequences in the previous section. For example, the binary sequence 2251111211, sketched in Figure 3, has the normalized autocorrelation function shown in Figure 4. Using the weighting suggested in equation (9) for $\ell = 2$ will modify this into a ternary sequence with the normalized autocorrelation function shown in Figure 5, which corresponds more to coloured noise.

Determination of the minimum time of experiment

We shall now consider the minimum value of the time of experiment, T, which is necessary in order that a good approximation to the desired power spectrum is obtained. As is well known from the uncertainty principle due to Heisenberg, there is a reciprocal relationship between the frequency spectrum of a signal and its duration in time. Hence, it is not possible to find a time function of finite duration which will correspond exactly to a prescribed frequency-limited signal. One can, therefore, only attempt to obtain a reasonable approximation in practical problems.

As proposed by Landau and Pollock [1961], the "time-spread", T, of f(t) may be defined by the equation

$$T^2 = \frac{\int_{-\infty}^{\infty} (t-t_o)^2 \; |f(t)|^2 \, dt}{\int_{-\infty}^{\infty} |f(t)|^2 \, dt} \tag{12}$$

and, correspondingly, the "frequency-spread" Ω of F(w) may be expressed by the equation

$$\Omega^2 = \frac{\int_{-\infty}^{\infty} (\omega-\omega_o)^2 \; |F(\omega)|^2 \, d\omega}{\int_{-\infty}^{\infty} |F(\omega)|^2 \, d\omega} \tag{13}$$

Then, for any choice of t_o and ω_o, it follows that

$$\Omega T \geq \frac{1}{2} \tag{14}$$

Since equation (14) provides only a rough approximation, the following has been proposed by MacCall in an unpublished paper (see Landau and Pollack, 1961)

$$\Omega T > 2\pi\alpha^2\beta^2 \tag{15}$$

where

$$\alpha = \frac{\int_{t_o}^{t_o+T} |f(t)|^2 \, dt}{\int_{-\infty}^{\infty} f(t)^2 \, dt} \tag{16}$$

and

$$\beta = \frac{\int_{\omega_o}^{\omega_o+\Omega} |F(\omega)|^2 \, d\omega}{\int_{-\infty}^{\infty} |F(\omega)|^2 \, d\omega} \tag{17}$$

For the case when $\beta^2 = 1$, the relationship between α^2 and ΩT has been calculated by Landau and Pollack [1961] and is shown in Figure 6. It will be seen that although initially α^2 increases linearly with ΩT, the curves indicate "saturation" with larger values of ΩT. Hence, it will be of very little practical value to make ΩT larger than, say, 5.

This, therefore, provides a simple rule for estimating the minimum reasonable duration of the experiment.

General procedure for designing the optimum staircase input

Based on the discussion in the previous sections, we may now describe the general procedure for designing the optimum staircase input of finite duration.

(i) Evaluate the function $|\Delta G(j\omega)|^2$ from the limiting values of the unknown parameters.

(ii) Truncating $|\Delta G(j\omega)|^2$ at some suitable upper frequency, determine the cut-off frequency ω_b.

(iii) The value of Ω^2 can now be evaluated using equation (13).

(iv) Hence, a suitable value of T is determined as $\frac{5}{\Omega}$.

(v) The sampling interval, t_s may be selected as half of the smallest time constant of the system. This may be estimated from the largest break-frequency (pole) of $G(j\omega)$ for various values of the parameters.

(vi) Hence, we get the number of samples, $N = \frac{T}{t_s}$

(vii) The staircase input signal may now be designed according to the procedure proposed in section 2, depending upon the nature of $|\Delta G(j\omega)|^2$.

CONCLUSIONS

A simple procedure has been proposed for designing an optimal staircase input of finite duration for the identification of linear systems. Digital implementation of the scheme is quite straightforward, and therefore, useful for practical applications.

REFERENCES

R.H. Barker, "Group Synchronizing of binary digital systems", Communication Systems", Communication Theory, London, pp. 273-287, 1953.

I. Burger, "A numerical method of process identification in automatic control systems", IFAC Symp. on Process Identification, (Prague, Czechoslovakia), June 1967.

D.A. Huffman, "The generation of impulse-equivalent pulse trains", IRE Trans. on Information Theory, vol. IT-8, Sept. 1962.

K. Izawa and K. Furute, "Apparatus to measure dynamic characteristics", 3rd IFAC Congress, London, 1966.

H. Kashiwagi, "A method of system identification using weighted m-sequence signal", Preprints 1974 JACC, pp. 285-292.

B. Kuzsta and N.K. Sinha, "Optimal input signals for system identification - a frequency domain approach", Proc. 18th Midwest Symp. on Circuits and Systems, Aug. 1975, pp. 149-152.

H.V. Landau and H.O. Pollack, "Prolate spheroid wave functions, Fourier analysis and uncertainty", Bell Syst. Tech. Journ., vol. ML, pp. 43-63, 1974.

P. Nikiforuk and M.M. Gupta, "A bibliography on the properties, generation and control system applications of shift register sequences", Int. Journal of Control, vol. 9, 1967, pp. 217-234.

R. Turyn, "Sequences with small correlation in error correcting codes", edited by H.B. Mann, John Wiley and Sons, 1968.

P.Y. Van Gerwen, "On the generation and application of pseudo-ternary codes in pulse transmission", Phillips Research Reports, vol. 20, pp. 469-489, 1965.

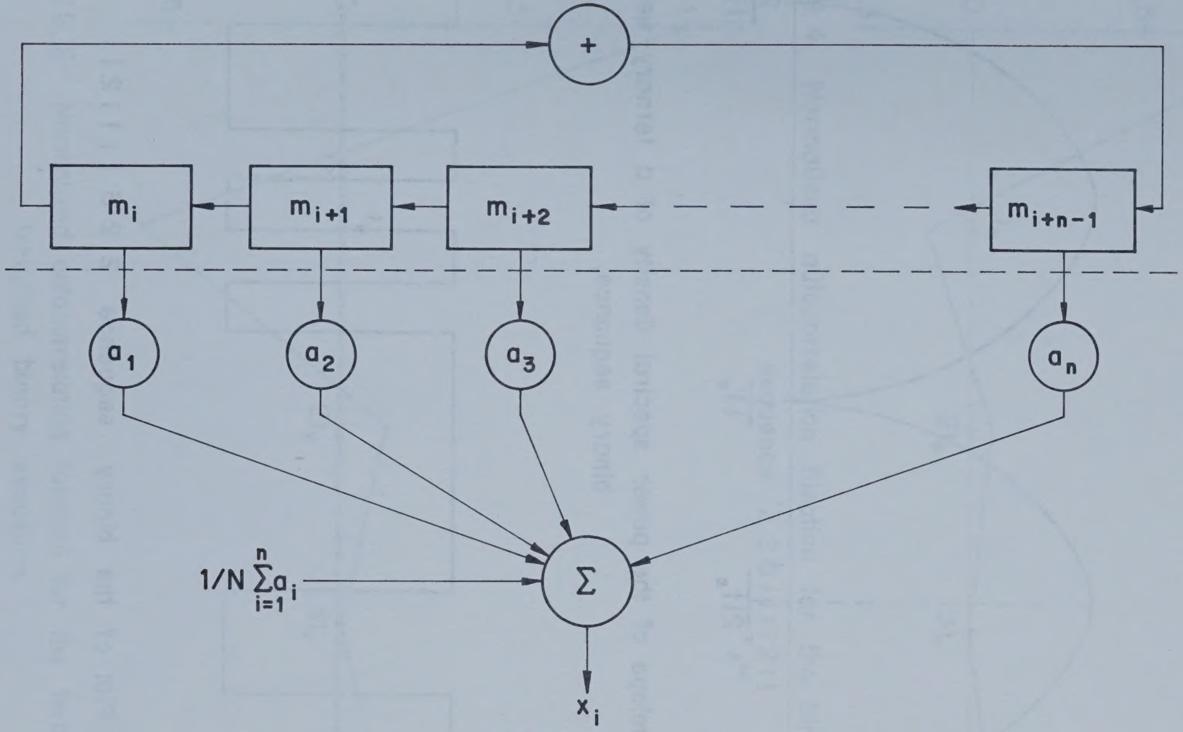

+
m_i
m_{i+1}
m_{i+2}
m_{i+n-1}
a_1
a_2
a_3
a_n
$1/N \sum_{i=1}^{n} a_i$
Σ
x_i

FIG. I A schematic diagram of the weighted m-sequence generator

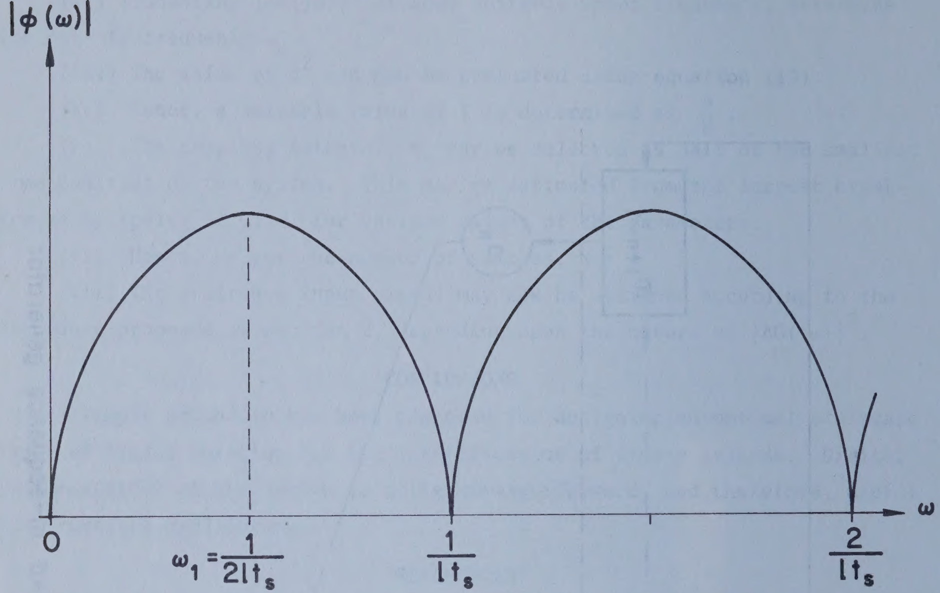


FIG. 2 Envelope of the power spectral density of a ternary-weighted binary sequence.

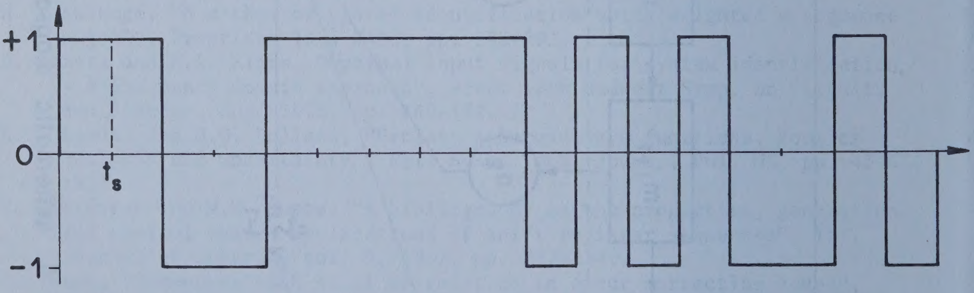


FIG. 3 Plot of the binary sequence 2 2 5 1 1 1 1 2 1 1

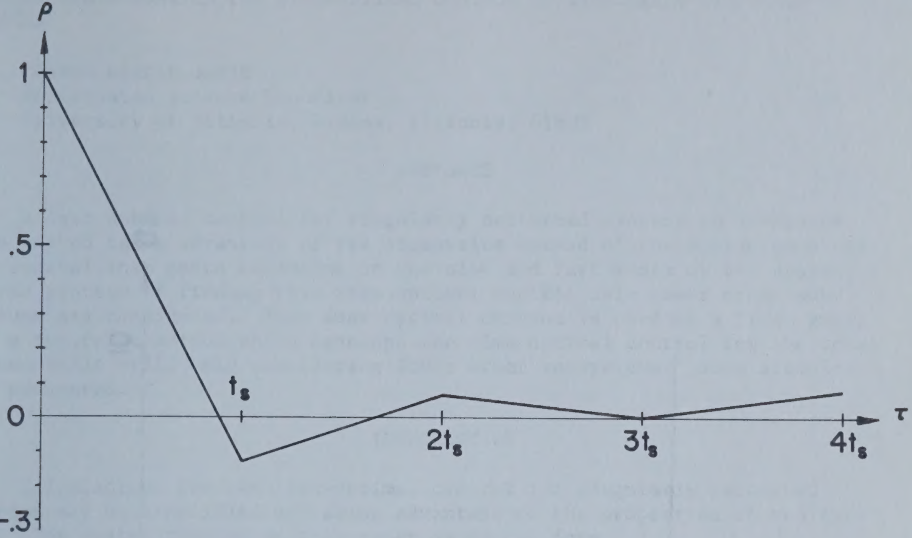


FIG. 4 Normalized autocorrelation function for the binary sequence 2 2 5 1 1 1 1 2 1 1

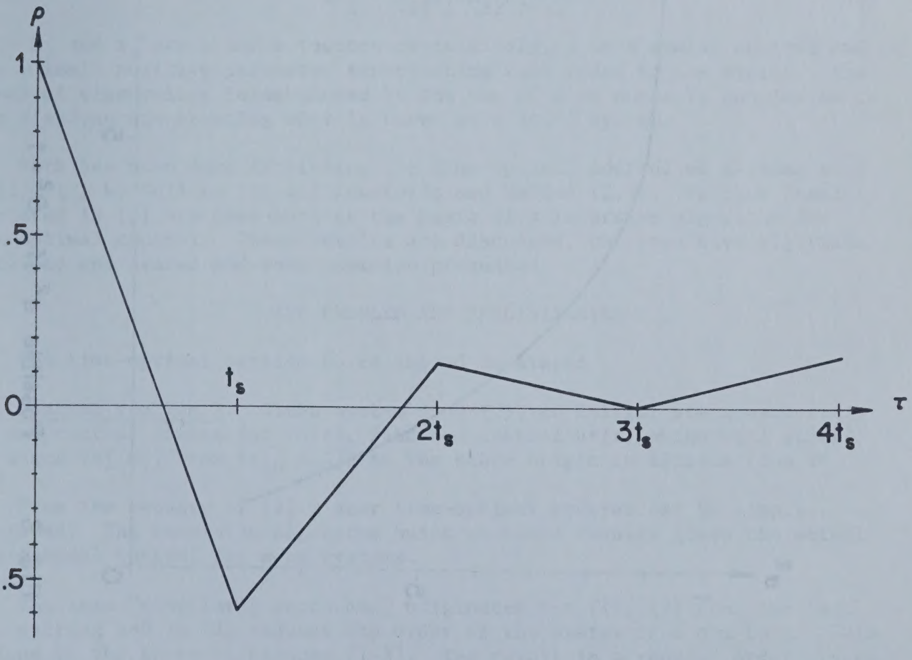


FIG. 5 Normalized autocorrelation function for the ternary-weighted binary sequence.

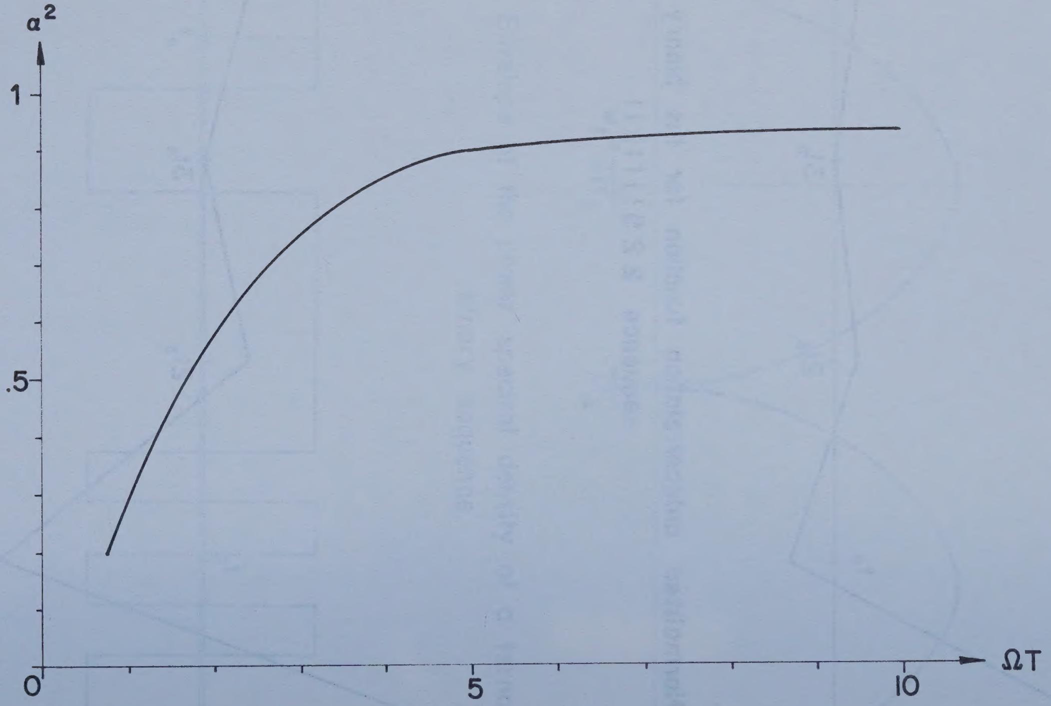


FIG. 6 Possible α^2 if $\beta^2 = 1$

A RECURSIVE ALGORITHM FOR TIME-OPTIMAL CONTROL OF SINGULARLY PERTURBED SYSTEMS

SHABON HAROLD JAVID
Coordinated Science Laboratory
University of Illinois, Urbana, Illinois, 61801

ABSTRACT

A near optimal control for singularly perturbed systems is developed. This method takes advantage of the eigenvalue spread of the system to break the control into parts dependent on the slow and fast modes of the system. In the process of finding this near-optimal control only lower order subsystems are considered. This near optimal control is used as a first guess for a recursive method which develops the time-optimal control for the total system while still only considering lower order subsystems. Some examples are presented.

INTRODUCTION

Calculations for the time-optimal control for singularly perturbed systems may be simplified by taking advantage of the properties of the system. The system treated in this paper is of the form

$$\dot{x}_1 = A_{11}x_1+A_{12}x_2+b_1u \tag{1}$$

$$\mu\dot{x}_2 = A_{21}x_1+A_{22}x_2+b_2u \tag{2}$$

where x_1 and x_2 are n and m vectors respectively, u is a scalar control and μ is a small positive parameter representing fast modes in the system. The spread of eigenvalues foreshadowed by the use of μ is normally considered to be a disadvantage creating what is known as a stiff system.

Work has been done in finding the time-optimal control of systems such as (1), (2) by Collins [1] and Kokotovic and Haddad [2,3]. Various results as stated in [2] are used here as the basis of a recursive algorithm for time-optimal control. These results are discussed, the recursive algorithm developed and stated and some examples presented.

THE PROBLEM AND PRELIMINARIES

The time-optimal problem to be solved is stated.

Control Problem 1: Given system (1), (2), an initial state vector x_{10}, x_{20} and control constraint $|u|\leq 1$, find the control $u^*(t)$ which will steer the state $[x_1' \; x_2']'$ from $[x_{10}' \; x_{20}']'$ to the state origin in minimum time T^*.

From the results of [2] a near time-optimal control can be simply developed. The recursive algorithm built on these results gives the actual time-optimal control for such systems.

The term "singularly perturbed" originates for (1), (2) from the fact that setting $\mu=0$ in (2) reduces the order of the system from n+m to n. This is done in the three references [1-3]. The result is a reduced order system

$$\dot{\bar{x}}_1 = A_o\bar{x}_1+b_o\bar{u} \tag{3}$$

where

$$A_o = A_{11} - A_{12}A_{22}^{-1}A_{21}, \quad b_o = b_1 - A_{12}A_{22}^{-1}b_2. \tag{4}$$

This is particularly useful as suggested in [3] for systems where A_{21} is zero and A_{22} may represent actuator dynamics. There, the modes that are actually to be controlled are represented in the matrix A_{11}.

The controllability of (1), (2) is necessary before continuing. In [2] system (1), (2) is transformed into a block diagonal form and as a result, the following theorem is stated.

Theorem 1: If A_{22}^{-1} exists and if

$$\text{rank}\,[b_o \vdots A_o b_o \vdots \cdots \vdots A_o^{n-1} b_o] = n \tag{5}$$

$$\text{rank}\,[b_2 \vdots A_{22} b_2 \vdots \cdots \vdots A_{22}^{m-1} b_2] = m \tag{6}$$

then there exists a $\mu^{*}>0$ such that the system (1),(2) is controllable for a all $\mu \in (0,\mu^{*}]$.

The system treated here is (1), (2) where the matrix A,

$$A = \begin{bmatrix} A_{11} & A_{12} \\ \frac{1}{\mu}A_{21} & \frac{1}{\mu}A_{22} \end{bmatrix} \tag{7}$$

has real, distinct, negative eigenvalues. This system is assumed controllable. As a result, there exists a nonsingular transformation [4],

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = PT \begin{bmatrix} \zeta \\ \eta \end{bmatrix} \tag{8}$$

where P is a matrix of the eigenvectors of A and T is a diagonal matrix which causes the coefficients of the control to be the negative of the eigenvalues of A. The result is (9), (10).

$$\dot{\underline{\zeta}} = \Lambda_1 \underline{\zeta} - \underline{\lambda}_1 u \tag{9}$$

$$\mu\dot{\underline{\eta}} = \Lambda_2 \underline{\eta} - \underline{\lambda}_2 u \tag{10}$$

This system is a modal form of (1), (2) where Λ,

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \frac{1}{\mu}\Lambda_2 \end{bmatrix} \tag{11}$$

is diagonal with the eigenvalues of A arranged in increasing order as elements. Similarly the vector $\underline{\lambda}$,

$$\underline{\lambda} = \begin{bmatrix} \underline{\lambda}_1 \\ \frac{1}{\mu}\underline{\lambda}_2 \end{bmatrix} \tag{12}$$

has the diagonal elements of Λ as elements.

Equations (9) and (10) are referred to as the slow and fast subsystems respectively since they contain the slow and fast modes of (1), (2) seperately. As μ becomes small we say the subsystems are more separated.

The initial state $[\underline{x}_{10}' \ \underline{x}_{20}']'$ for the Control Problem 1 is transformed to $[\underline{\zeta}_o' \ \underline{\eta}_o']'$ by the relation

$$\begin{bmatrix} \underline{\zeta}_o \\ \underline{\eta}_o \end{bmatrix} = T^{-1}P^{-1} \begin{bmatrix} \underline{x}_{10} \\ \underline{x}_{20} \end{bmatrix} . \tag{13}$$

In Control Problem 1 the desire is to steer the state $[x_1' \ x_2']$ to the origin. Clearly steering $[\underline{\zeta}' \ \eta']'$ to a final state at the origin results in $[x_1' \ x_2']'$ being at the origin. The transformed control problem may be stated.

Control Problem 2: Given (9), (10) with u constrained $|u|\leq 1$, and initial state $[\zeta_0' \ \eta_0']'$, find the control u*(t) which steers the state of (9), (10) from the initial state to the origin in minimum time T*.

From the minimum principle the time-optimal control for this system is

$$u^*(t) = -\mathrm{sgn}\{\underline{\lambda}_1' e^{\Lambda_1'(T^*-t)}p_f + \frac{\underline{\lambda}_2'}{\mu} e^{\Lambda_2'\left(\frac{T^*-t}{\mu}\right)} q_f\} \tag{14}$$

where $[p_f' \ q_f']'$ is the value of the costate vector $[p'q']$ at time $t=T^*$. An analysis similar to that in [2] gives us the following:

1. The control u*(t) may be thought of as the concatenation of two controls $u_s^*(t)$ and $u_f^*(t)$, where $u_s^*(t)$ is a control dependent primarily on subsystem (9) and $u_f^*(t)$ is a control primarily dependent on subsystem (10).

2. A near time-optimal control $\hat{u}(t)$ (defined in the next section) can be found such that it has the same control sequence (sequence of plus and minus ones) as u*(t) and steers $[\underline{\zeta}'\underline{\eta}']'$ to an $0(\mu)$ neighborhood of the origin in near minimum time $\hat{T}$.

3. It may be possible, by perturbing the switching times of $\hat{u}(t)$ to arrive at u*(t).

THE RECURSIVE ALGORITHM

Suppose, for the moment, that we know for a particular problem, the optimal control u*(t). It is generally known that this control will be made of a sequence of constant values equal to ±1. Each constant value is applied for a period of time t_i, where $i=1,2,\ldots,n+m$. Two examples of such a control for a system in which n=2 and m=2 are (15) and (16).

$$u^*(t) = \begin{cases} +1 & t \in [0,t_1) \\ -1 & t \in [t_1, t_1+t_2) \\ +1 & t \in [t_1+t_2, t_1+t_2+t_3) \\ -1 & t \in [t_1+t_2+t_3, t_1+t_2+t_3+t_4) \end{cases} \tag{15}$$

$$u^*(t) = \begin{cases} +1 & t \in [0, t_1) \\ -1 & t \in [t_1 \ t_1+t_2) \\ -1 & t \in [t_1+t_2, t_1+t_2+t_3) \\ +1 & t \in [t_1+t_2+t_3, t_1+t_2+t_3+t_4) \end{cases} \tag{16}$$

In both cases $t_1+t_2+t_3+t_4=T^*$. The time t_I^*, where

$$t_I^* = t_1+t_2 \tag{17}$$

is defined as the intermediate time. From [2] we know that the control prior to t_I^* is dependent primarily on the modes of the slow subsystem and the control after t_I^* is dependent primarily on the modes of the fast subsystem. However [2] does not define exactly how to determine the optimal control. Let $[\underline{\zeta}'\underline{\eta}']'$ represent the solution to (9) and (10) with an optimal control input $u^*(t)$. Then the point $[\underline{\zeta}^*(t_I^*)'\underline{\eta}^*(t_I^*)']'$ may be defined as the intermediate point.

A near time-optimal control $\hat{u}(t)$ can be found by the following method:

The control $\hat{u}_s(t)$ is found which will steer the state of the slow subsystem from $\underline{\zeta}_0$ to the origin in minimum time $\hat{t}_I$. This defines a fast intermediate state $\underline{\eta}(\hat{t}_I)$ resulting from the application of $\hat{u}_s(t)$ to (10). The control $\hat{u}_f(t)$ is found which will steer the state of the fast subsystem (10) from $\underline{\eta}(\hat{t}_I)$ to the origin in minimum time. The control $\hat{u}(t)$ is made of $\hat{u}_s(t)$ concatenated with $\hat{u}_f(t)$. The total time required for $\hat{u}(t)$ is $\hat{T}$. This implies that $\hat{u}_f(t)$ is applied for time $\hat{T}-\hat{t}_I$.

Careful study of equation (14) allows us to state:

Theorem 2: Given system (9), (10) with an initial state $[\underline{\zeta}_0'\underline{\eta}_0']'$ there exists a $\mu^*>0$ such that if $\mu \in (0,\mu^*)$, the control $\hat{u}(t)$ is near time-optimal in the sense that steers the state of (9), (10) and hence of (1), (2) from an initial state to an $O(\mu)$ neighborhood of the origin in near minimum time $\hat{T}$. The control sequence of $\hat{u}(t)$ is optimal (the same as $u^*(t)$). Also the time $\hat{T}-\hat{t}_I$ is $O(\mu)$ relative to $\hat{t}_I$. When $\mu \in (0,\mu^*)$ the system is said to be sufficiently separated.

It is important to note that due to the control constraint μ^* is dependent on the initial state. The fact that $\hat{u}(t)$ has the optimal control sequence and $\hat{T}$ is near T^* suggests that some perturbations of the switching times of $\hat{u}(t)$ will produce the optimal control $u^*(t)$. The recursive algorithm presented here is designed to do just this and is essentially a successive approximation for the intermediate point. From the value of the intermediate point the control $u^*(t)$ may be found while considering the subsystems separately. Through this indirect means a perturbation of the switching times is achieved.

The first guess for this point is taken to be

$$\begin{bmatrix} \underline{\zeta}^o \\ \underline{\eta}^o \end{bmatrix} = \begin{bmatrix} 0 \\ \underline{\eta}(\hat{t}_I) \end{bmatrix} \tag{18}$$

as described. The control $u^0(t)$, the first guess for $u^*(t)$, is defined as the control $\hat{u}(t)$,

$$u^0(t) = \hat{u}(t) \tag{19}$$

where

$$u_s^0(t) = \hat{u}_s(t) \tag{20}$$

and

$$u_f^0(t) = \hat{u}_f(t) \tag{21}$$

From the control $u^0(t)$ the optimal control sequence is found. Since we know this sequence is optimal, for μ small enough, it is used in determining the controls $u^n(t)$ for the remainder of the algorithm.

We know that $\underline{\eta}(T^0)$ where

$$T^0 = \hat{T}\ , \tag{22}$$

is a zero vector from the definition of $u_f^{\,0}(t)$ and also that $\underline{\zeta}(T^0)$ is something other than a zero vector. Assume for the moment, that $u_f^0(t)$ is the optimal control $u_f*(t)$, the control that follows time t_I*. We would need only to find the value of the slow state, $\underline{\zeta}^1$ from which the slow state will be steered to the origin by $u_f^0(t)$. If $u_f^0(t)$ had actually been optimal a control $u_s^{\,1}(t)$ which would steer the slow state from $\underline{\zeta}_0$ to $\underline{\zeta}^1$ would be found and the optimal control would then be $u_s^{\,1}(t)$ concatenated with $u_f^{\,0}(t)$. However $u_f^0(t)$ is most likely not optimal, but the guess $\underline{\zeta}^1$ is used to begin the next iteration of the algorithm presented in Figure 1. The steps of this algorithm are described in more detail as follows:

1. This is the step that initiates the entire process.
2. This sets the iteration number to zero.
3. The control $u_s^{\,0}(t)$ steers $\underline{\zeta}(t)$ from $\underline{\zeta}_0$ to $\underline{\zeta}^0$, the origin, in minimum time.
4. The state $\underline{\eta}(t)$ is steered to $\underline{\eta}^0$ from $\underline{\eta}_0$ by $u^0(t)$.
5. The control $u_f^0(t)$ steers $\underline{\eta}(t)$ from the fast intermediate point $\underline{\eta}^0$ to the origin in minimum time.
6. The test for separation decides whether $T^0 - t_I^{\,0}$ is $0(\mu)$ relative to $t_I^{\,0}$. If this is satisfied the algorithm has experimentally worked will as a computer program. There is as yet no proof that this is the necessary condition for separation.
7. The optimal control sequence is that of $u_s^{\,0}(t)$ concatenated with that of $u_f^{\,0}(t)$. This sequence is used for the remainder of the algorithm.
8. The point $\underline{\zeta}^{n+1}$, the next guess for the slow state's portion of the intermediate point, is found by integrating equation (9) backwards from the origin with u as the control $u_f^{\,n}(t)$. This is the crux of the recursive algorithm.
9. Increment the iteration number by one.
10. The control $u_s^{\,n}(t)$ is the control with the control sequence of $u_s^{\,0}(t)$ that steers $\underline{\zeta}(t)$ from $\underline{\zeta}_0$ to $\underline{\zeta}^n$.
11. The fast state $\underline{\eta}(t)$ is steered from $\underline{\eta}_0$ to $\underline{\eta}^n$ by $u_s^{\,n}(t)$.
12. The control $u_f^{\,n}(t)$ has the control sequence of $u_f^{\,0}(t)$ and steers $\underline{\eta}(t)$ from $\underline{\eta}^n$ to the origin.
13. The time $T^n$is the period required for the control $u^n(t)$. The vector $\underline{\zeta}(T^n)$ is the value of the slow state resulting from the application of $u^n(t)$ to the slow subsystem (9) at time $t_0=0$. By definition $\underline{\eta}(T^n)$ is $\underline{0}$ Therefore, if $\underline{\zeta}(T^n)$ is near enough to the origin the desired control has been found.
14. In the actual use of the algorithm with systems and initial conditions which satisfy the test for separation of step 6, ten iterations are all that is necessary to find this desired control $u*(t)$ to five or six places. Therefore the algorithm is terminated after ten iterations.

OPTIMALITY

We know from [2] that the optimal control $u*(t)$ has the control sequence of $u^0(t)$. Also some perturbation of the switching times will result in $u^0(t)$ becoming $u*(t)$. This is actually what the recursive algorithm accomplishes, a perturbation of the switching times indirectly through a sequence of times $t_I^{\,n}$ and points $[\underline{\zeta}^{n\prime}\ \underline{\eta}^{n\prime}]'$which converge to t_I* and $[\underline{\zeta}*'(t_I*)\ \eta*'(t_I*)]'$.

EXAMPLES

Example 1: This example demonstrates that the recursive algorithm is capable of finding a control for system (9), (10) with eigenvalues -.5, -1., -5., -10.2, that requires only two switchings. This control was chosen in advance

$$u(t) = \begin{cases} +1 & t \in [0, .2) \\ -1 & t \in [.2, .3) \\ +1 & t \in [.3, .31). \end{cases} \tag{23}$$

and then initial condition (24) was calculated from the control.

$$\begin{bmatrix} \zeta 10 \\ \zeta 20 \\ \eta 10 \\ \eta 20 \end{bmatrix} = \begin{bmatrix} -\ .054331312 \\ -\ .106513015 \\ -\ .184655697 \\ -4.656111629 \end{bmatrix} \tag{24}$$

In order for the algorithm to succeed we would expect the intermediate point to occur at a time between 0.2 and 0.3 seconds. This would imply that $-u_1 = u_2$. This is precisely what occurs.

At the end of the first iteration the control $u^0(t)$ has the optimal control sequence and steers the state of system (9), (10) to within a 10^{-3} neighborhood of the origin.

$$u^0(t) = \begin{cases} +1 & t \in [0, .199) \\ -1 & t \in [.199, .289) \\ -1 & t \in [.289, .296) \\ +1 & t \in [.296, .304) \end{cases} \tag{25}$$

After nine iterations the control $u^9(t)$ is found which steers the state of (9), (10) to within a 10^{-6} neighborhood of the origin.

$$u^9(t) = \begin{cases} +1 & t \in [0, .199999) \\ -1 & t \in [.199999, .2896448) \\ -1 & t \in [.2896448, .2999974) \\ +1 & t \in [.2999974, .30999624). \end{cases} \tag{26}$$

The intermediate time t_I^9 is .2896448 for $u^9(t)$. This control (26) is accurate to five places. By running the iterative method for more iterations more accuracy is obtained.

How about separation? Here

$$t_1^9 + t_2^9 \cong .29$$
$$\text{and } t_3^9 + t_4^9 \cong .020. \tag{27}$$

Thus, the control time primarily dependent on the fast eigenvalues is approximately 6.9% of the control time dependent on the slow eigenvalues. This is certainly $O(\mu)$ and we may say the system is sufficiently separated.

Example 2: In this example the recursive method is applied to a sixth order system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \mu\dot{x}_3 \\ \mu\dot{x}_4 \\ \mu\dot{x}_5 \\ \mu\dot{x}_6 \end{bmatrix} = \begin{bmatrix} -0.05 & & & & & \\ & -.1 & & & 0 & \\ & & -.05 & & & \\ & 0 & & -.1 & & \\ & & & & -.5 & \\ & & & & & -1.02 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0.05 \\ .1 \\ .05 \\ .1 \\ .5 \\ 1.02 \end{bmatrix} u \tag{28}$$

The control $u_o(t)$

$$u_o(t) = \begin{cases} +1 & t \in [0, 52) \\ -1 & t \in [52., 58.8) \\ +1 & t \in [58.8, 60.9) \\ -1 & t \in [60.9, 61.46) \\ +1 & t \in [61.46, 61.99) \\ -1 & t \in [61.66, 61.77) \end{cases} \tag{29}$$

is found which steers the state of (28) from initial condition (30) to an $O(\mu)$ neighborhood of the origin

$$\begin{bmatrix} x_{10} \\ x_{20} \\ x_{30} \\ x_{40} \\ x_{50} \\ x_{60} \end{bmatrix} = \begin{bmatrix} -7 \\ -3 \\ -5.2 \\ -8.9 \\ -15 \\ -2.234 \end{bmatrix} \tag{30}$$

The intermediate time $t_I^{(0)}$ is

$$t_I^{(0)} = 58.8 \text{ seconds.}$$

After ten iterations of the procedure the state of the system was steered from initial condition (30) to within a 10^{-13} neighborhood of the origin by the control

$$u^{(10)}(t) = u^*(t) = \begin{cases} 1 & t \in [0, 52.35) \\ -1 & t \in [52.35, 61.03) \\ 1 & t \in [61.03, 63.22) \\ -1 & t \in [63.22, 63.98) \\ 1 & t \in [63.98, 64.19) \\ -1 & t \in [64.19, 64.25) \end{cases} \tag{31}$$

CONCLUSION

The control developed for system (9), (10) by the recursive algorithm the control desired for system (1), (2). Experimentally as μ becomes smaller the convergence of the method becomes more rapid taking advantage of the stiffness of the systems.

REFERENCES

1 Collins, W. D. "Singular Perturbations of Linear Time-Optimal Control Problems", Recent Mathematical Developments in Control, D. J. Bell, Ed. New York: Academic, 1973.

2 Kokotovic P. K., Haddad A. H., "Controllability and Time-Optimal Control of Systems with Slow and Fast Modes", IEEE Trans. on Automatic Control, Vol. AC-20, No. 1, February 1975, pp. 111-113.

3 Kokotovic P. K., Haddad A. H., "Singular Perturbations of a Class of Time-Optimal Controls", IEEE Trans. on Automatic Control, Vol. AC-20, No. 1, February 1975, pp. 163-164.

4 Javid, S. H. "The Time-Optimal Control of a Class of Singularly Perturbed Systems", MS thesis, Dept. Elect. Eng., University of Illinois, 1975.

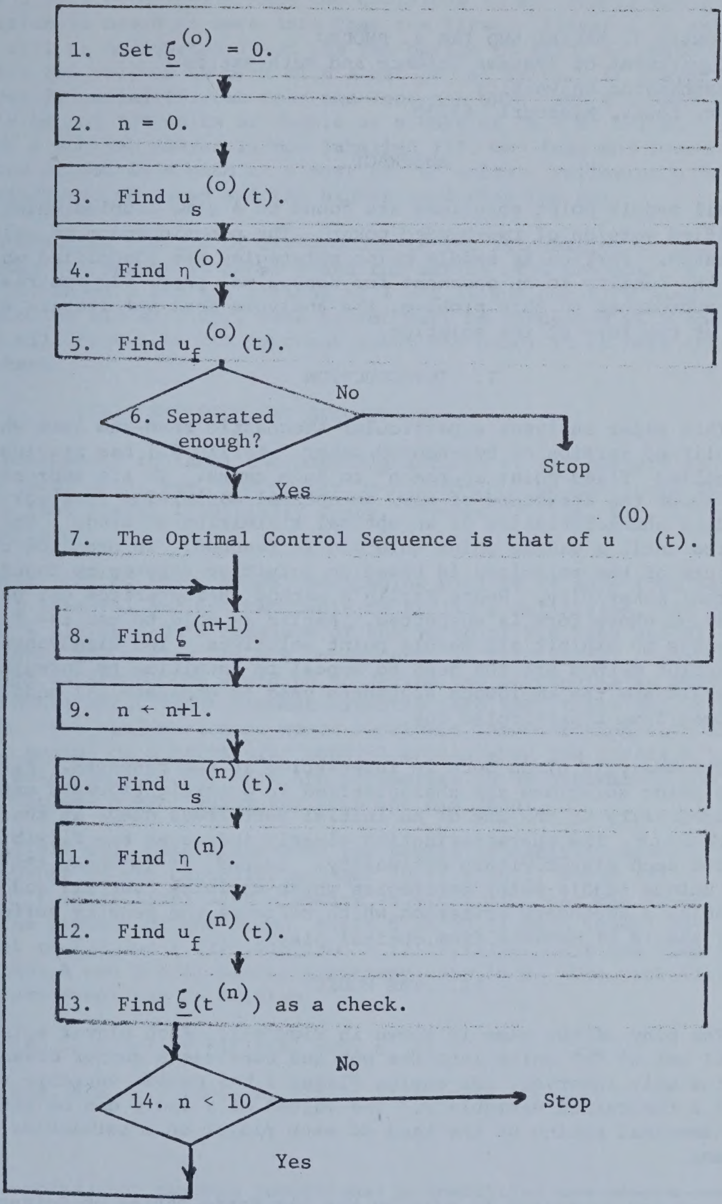

1. Set $\underline{\zeta}^{(o)} = \underline{0}$.
2. n = 0.
3. Find $u_s^{(o)}(t)$.
4. Find $\eta^{(o)}$.
5. Find $u_f^{(o)}(t)$.
6. Separated enough?
No
Stop
Yes
7. The Optimal Control Sequence is that of $u^{(0)}(t)$.
8. Find $\underline{\zeta}^{(n+1)}$.
9. $n \leftarrow n+1$.
10. Find $u_s^{(n)}(t)$.
11. Find $\underline{\eta}^{(n)}$.
12. Find $u_f^{(n)}(t)$.
13. Find $\underline{\zeta}(t^{(n)})$ as a check.
No
14. $n < 10$
Stop
Yes

Figure 1. The steps in the Iterative Method.

OPTIMAL STRATEGIES IN SIMPLIFIED TWO-PLAYER POKER

DONALD J. HALSKI AND IAN B. RHODES
Department of Systems Science and Mathematics
Washington University
St. Louis, Missouri 63130

ABSTRACT

All saddle point solutions are found to a game problem which models a simplified version of two-handed poker. The nonuniqueness of solutions is delineated. Particular saddle point strategies are identified which maximize the penalty to an opponent for nonoptimal play. In contrast to previous approaches to this problem, the analysis does not require an initial guess at the form of the solution.

I. INTRODUCTION

This paper analyzes a particular stochastic zero-sum game which models a simplified version of two-handed poker. Karlin [1] has previously applied a so-called "fixed-point approach" to such games. In his approach, assumptions about the structure of say, an optimal maximizing strategy are used to deduce characteristics of an optimal minimizing strategy. The process is iterated until a saddle point solution is reached. In practice the assumed structure of the solutions is based on intuitive strategies found effective in actual poker play. Hence Karlin's method characterizes the parameters of a solution whose form is suspected. Karlin is able to use the resulting strategies to exhibit all saddle point solutions. Two disadvantages of the fixed-point method are the need to appeal to intuition to formulate an initial guess and the ambiguous arguments used to generate all saddle point solutions from a particular one.

The analysis given here is interesting in two respects. First, all saddle point solutions are characterized in a straightforward manner without circularity or the use of an initial fortuitous guess at the form of the solutions. The characterization clearly indicates the flexibility afforded each player within optimality. Second, it is shown that there exist unique saddle point strategies which maximize over all saddle point strategies a secondary criterion which reflects the penalty suffered by a player should he deviate from optimal play.

II. THE MODEL

The play of the game is shown in Figure 1. Each player enters an initial bet of "k" units into the pot and receives a number drawn randomly from the unit interval. We assign Player 1 the random variable x and Player 2 the random variable y. The values of x and y can be interpreted as a numerical rating of the hand of each player on a particular play of the game.

This work was supported in part by the Air Force Office of Scientific Research, Air Force Systems Command, under Grant AFOSR-73-2496.

Player 1 will potentially make two decisions in any sample game; the second decision is based on more data than the first. Player 1 on the first move will be designated I and has the option to fold, in which case Player 2 wins the pot, or to open with a bet of "a" units. If Player 1 opens, Player 2, called II, can fold and lose his ante, see I's hand by matching I's bet of "a" units or double at a cost of "a + b" units. In the event of a double, Player 1, now labelled III, may fold and lose "k + a" units or see II's hand at a cost of "b" units. Following a "see" response, hands are compared and the higher hand wins the pot.

The interesting feature of the problem is its information structure. Player I knows x. Player II knows y and the action of I but not I's hand x. Player III knows x and the actions of I and II but not II's hand y. Thus, in choosing his strategy, each player must be concerned with what his actions will reveal to his opponent about his hand; it is here that bluffing enters.

III. NOTATION AND ASSUMPTIONS

It is convenient to label the possible actions of the players as F, O, S, D for fold, open, see, and double, respectively. We designate by IO the event that Player 1 initially opens. Events IF, IIS, etc. are similarly defined.

Our definition of the players' strategies will be as maps from data into the unit interval and is comparable to Karlin's definition. In this formulation, a player's data consists of the value of his hand and all previous decisions; the strategies are interpreted as conditional probabilities. An alternative scheme would be to define the strategies as maps from data and an additional random element directly into the finite set of available actions. The random element would represent randomization such as a coin flip to decide on a particular control action when the strategy is mixed. The weightings of the coin are parameters to be determined and could depend on the player using the coin and his data.

The analysis will take place over the sample probability space. Let the set of events be the Cartesian product $[0,1] \times [0,1] \times \{IF, IO\}$ $\{IIF, IIS, IID\} \times \{IIIF, IIIS\}$ with elements (x,y,c,d,e). Let the associated σ-field be the product of the Borel field $\mathcal{B}$ on $[0,1]$ with itself and with the power set of $\{IF, IO\} \times \{IIF, IIS, IID\} \times \{IIIF, IIIS\}$. For the remainder of this section A and B will denote arbitrary sets in $\mathcal{B}$. The probability of an arbitrary event can be written

$$P[x\varepsilon A,\ y\varepsilon B,\ c,\ d,\ e] = P[e|x\varepsilon A,\ y\varepsilon B,\ c,\ d] \cdot$$

$$P[d|x\varepsilon A,\ y\varepsilon B,\ c] \cdot P[c|x\varepsilon A,\ y\varepsilon B] \cdot P[y\varepsilon B|x\varepsilon A] \cdot P[x\varepsilon A] \quad .$$

It should be noted that certain conditional probabilites are always zero, reflecting the conclusion of the game after a fold or a see response. For example, $P[IIS|x\varepsilon A,\ y\varepsilon B,\ IF] = 0$.

The strategies of the players are defined by the pointwise values of regular conditional probabilities which can be shown to exist by the regularity theorem in Loeve [2, p.363]. The conditioning elements indicate the maximal information available during the course of play when strategy values must be chosen. Players I and III each have complementary options and so a single conditional probability suffices to specify the strategies of each. Player II has three options and requires two conditional probabilities to describe his strategy. The strategies are defined by

$$\xi_1(x) = P[IO \mid x] \tag{1}$$

$$\xi_2(x) = P[IIIS \mid x,\ IO,\ IID] \tag{2}$$

$$\eta_1(y) = P[IIS \mid y,\ IO] \tag{3}$$

$$\eta_2(y) = P[IID \mid y,\ IO] \quad . \tag{4}$$

It is important to realize that for an arbitrary selection of feasible strategies the conditional probabilities (1) to (4) are functionally independent of an opponent's hand. Since we will subsequently assume that the values of x and y are chosen independently, the probability of an arbitrary event may be simplified to

$$P[x\varepsilon A,\ y\varepsilon B,\ c,\ d,\ e] = P[e \mid x\varepsilon A,\ c,\ d] \cdot P[d \mid y\varepsilon B,\ c] \cdot \tag{5}$$

$$P[c \mid x\varepsilon A]\ P[y\varepsilon B] \cdot P[x\varepsilon A] \quad .$$

The following assumptions will be made throughout the paper and are identical to those used by Karlin [1].

$$k,\ a,\ b \text{ are strictly positive} \quad . \tag{6}$$

The random variables x and y are independent and uniformly distributed over the unit interval. (7)

The uniform distribution may be replaced by one absolutely continuous with respect to Lebesgue measure without altering the method of analysis or the form of the resulting strategies. The absolute continuity assumption implies that we need only characterize the optimal strategies almost everywhere with respect to Lebesgue measure and that we may ignore events of zero probability such as the equality of hands.

III. PROBLEM STATEMENT

The value of the game gives the net winnings of Player 1 and takes the following equivalent forms:

$$K(\xi_1,\ \xi_2,\ \eta_1,\ \eta_2) = -k + \int_0^1 [\xi_1(x)M_1(x) + \xi_1(x)\xi_2(x)M_2(x)]dx \tag{8}$$

$$= -k + 2j \int_0^1 \xi_1(x)dx \tag{9}$$

$$+ \int_0^1 [\eta_1(y)N_1(y) + \eta_2(y)N_2(y)]dy$$

$$\text{where}\quad M_1(x) = 2k - (2k+a)\int_x^1 \eta_1(y)\,dy + a\int_0^x \eta_1(y)\,dy \tag{10}$$

$$- (2k+a)\int_0^1 \eta_2(y)\,dy$$

$$M_2(x) = (2k+2a+b)\int_0^x \eta_2(y)\,dy - b\int_x^1 \eta_2(y)\,dy \tag{11}$$

$$N_1(y) = -(2k+a)\int_0^y \xi_1(x)\,dx + a\int_y^1 \xi_1(x)\,dx \tag{12}$$

$$N_2(y) = -(2k+a)\int_0^1 \xi_1(x)\,dx - b\int_0^y \xi_1(x)\xi_2(x)\,dx \tag{13}$$

$$+ (2k+2a+b)\int_y^1 \xi_1(x)\;\xi_2(x)\,dx \;.$$

Our simplified two-player poker game is reduced to the following mathematical problem.

Problem 1: Characterize all solutions, if any exist, to the saddle point condition

$$K(\xi_1, \xi_2, \overset{*}{\eta}_1, \overset{*}{\eta}_2) \le K(\overset{*}{\xi}_1, \overset{*}{\xi}_2, \overset{*}{\eta}_1, \overset{*}{\eta}_2) \le K(\overset{*}{\xi}_1, \overset{*}{\xi}_2, \eta_1, \eta_2) \tag{14}$$

subject to the following constraints

$$0 \le \xi_j(x) \le 1 \quad , \quad j = 1,2 \quad , \quad x\varepsilon[0,1] \tag{15}$$

$$0 \le \eta_j(y) \quad , \quad j = 1,2 \quad , \quad y\varepsilon[0,1] \tag{16}$$

$$0 \le \eta_1(y) + \eta_2(y) \le 1 \quad , \quad y\varepsilon[0,1] \;. \tag{17}$$

IV. RESULTS

In this section we give the principal results in the form of three theorems and an associated figure. Proofs of the theorems can be found in the appendix.

Theorem 1. The set of strategies $(\overset{*}{\xi}_1, \overset{*}{\xi}_2, \overset{*}{\eta}_1, \overset{*}{\eta}_2)$ is a solution to Problem 1 if and only if the following conditions are satisfied.

$$\text{Parameters:}\quad \int_\beta^1 dx = \frac{2k(k+a)(2k+2a+b)}{(2k+a)(k+a+b)(4k+4a+b)} \tag{18}$$

$$\int_{\gamma}^{1} dx = \frac{(2k+2a+b)}{(k+a)} \int_{\beta}^{1} dx \qquad (19)$$

Player 1: $$\xi_1^*(x) = 1 \;, \qquad x \varepsilon (\gamma,1] \qquad (20)$$

$$\xi_2^*(x) = \begin{cases} 0 \;, & x \varepsilon [0,\gamma) \\ 1 \;, & x \varepsilon (\beta,1] \end{cases} \qquad (21)$$

$$\int_{0}^{\gamma} \xi_1^*(x)\,dx = \frac{a}{2k+a} \int_{\gamma}^{1} dx \qquad (22)$$

$$\int_{\gamma}^{\beta} \xi_2^*(x)\,dx = \int_{\beta}^{1} dx \qquad (23)$$

$$\frac{k+a}{k+a+b} \int_{x}^{\beta} dz \leq \int_{x}^{\beta} \xi_2^*(z)\,dz \;, \quad x \varepsilon (\gamma,\beta) \qquad (24)$$

Player 2: $$\eta_1^*(y) = \begin{cases} 0 \;, & y \varepsilon [0,\gamma) \cup (\beta,1] \\ 1 \;, & y \varepsilon (\gamma,\beta) \end{cases} \qquad (25)$$

$$\eta_2^*(y) = \begin{cases} 0 \;, & y \varepsilon (\gamma,\beta) \\ 1 \;, & y \varepsilon (\beta,1] \end{cases} \qquad (26)$$

$$\int_{0}^{\gamma} \eta_2^*(y)\,dy = \frac{b}{2k+2a+b} \int_{\beta}^{1} dx \;. \qquad (27)$$

Figure 2 pictorially describes a saddle point solution and the corresponding functions $M_i(\cdot)$, $N_i(\cdot)$, $i=1,2$. The value of the game can be computed from Theorem 1 and shows a value in favor of Player 2.

Corollary. Let K^* be the value of the game at a saddle point. Then

$$K^* = -k \left[1 - \frac{4k(k+a)(2k+2a+b)^2}{(2k+a)^2(k+a+b)(4k+4a+b)} \right] \qquad (28)$$

and $$-k < K^* < 0 \;. \qquad (29)$$

The existence of at least one set of saddle point strategies is guaranteed by Theorem 2.

Theorem 2. The following strategies satisfy (18) through (27).

Parameters: γ and β are given by (18) and (19).

$$\int_{\varepsilon}^{1} dx = 2 \int_{\beta}^{1} dx \qquad (30)$$

$$\int_{\mu}^{1} dx = \frac{(4k+4a+b)(k+a+b)}{(2k+2a+b)(k+a)} \int_{\beta}^{1} dx \tag{31}$$

$$\int_{\delta}^{1} dx = \frac{2(2k+2a+b)}{2k+a} \int_{\beta}^{1} dx \tag{32}$$

$$\text{Player 1:} \quad \xi_1^*(x) = \begin{cases} 0 \quad , & x\varepsilon[0,\delta) \\ 1 \quad , & x\varepsilon(\delta,1] \end{cases} \tag{33}$$

$$\xi_2^*(x) = \begin{cases} 0 \quad , & x\varepsilon[0,\varepsilon) \\ 1 \quad , & x\varepsilon(\varepsilon,1] \end{cases} \tag{34}$$

$$\text{Player 2} \quad \xi_1^*(y) = \begin{cases} 0 \quad , & y\varepsilon[0,\gamma) \cup (\beta,1] \\ 1 \quad , & y\varepsilon(\gamma,\beta) \end{cases} \tag{35}$$

$$\xi_2^*(y) = \begin{cases} 0 \quad , & y\varepsilon[0,\mu) \cup (\gamma,\beta) \\ 1 \quad , & y\varepsilon(\mu,\gamma) \cup (\beta,1] \end{cases} \tag{36}$$

The noteworthy aspect of Theorem 1 is not so much that the saddle point strategies are nonunique but that they can all be characterized. The flexibility of optimal play is clearly delineated by (22), (23), (24) and (27). We can now pose the question of the existence of a secondary criterion which will cause each player to choose a particular saddle point strategy from all those available to him. The answer is provided in Theorem 3; it is interesting that the resulting strategies are pure.

Theorem 3. The strategies of Theorem 2 maximize uniquely over all saddle point strategies the penalty to an opponent for violating the conditions of Theorem 1.

V. DISCUSSION

The nonuniqueness of saddle point strategies can be given intuitive motivation. For a hand from $[0,\gamma)$, Player I chooses between open and fold according to any distribution satisfying (22). If $y > \gamma$, I always loses no matter what the form of $\xi_1^*(\cdot)$ on $[0,\gamma)$. Should it happen that $y < \gamma$, Player I is willing to gamble an amount "a" on the event that $x > y$ or that II will fold rather than double; the probability of this event is independent of $\xi_1^*(\cdot)$ on $[0,\gamma)$. Similarly, given the opportunity to play, Player II may double or fold on $[0,\gamma)$ with any distribution satisfying (27). Either $x\varepsilon[\gamma,1]$ and III will see with a fixed probability independent of the doubling strategy $\eta_2^*(\cdot)$ on $[0,\gamma)$ and will win; or $x\varepsilon[0,\gamma)$ and III will always fold and II will win. Finally, in the event of a double, Player III deduces that $y\varepsilon[0,\gamma) \cup (\beta,1]$. For $x\varepsilon(\gamma,\beta)$ III may use any distribution to decide between see and fold modulo the constraints (23) and (24). Either $y < \gamma$ and III always wins or $y > \beta$ and III always loses.

Karlin [1] guessed the correct form of the bluffing strategies of I and II at the outset of his fixed-point analysis based on "intuitive considerations." The flexibility afforded III is discovered only after a circuitous discussion of the generality of his proposed solution. This is in contrast to our proof of the necessity of Theorem 1 which consists solely of a series of consequences when each player acts optimally. The proofs of sufficiency of Theorem 1 and the subsequent results are algebraic.

CONCLUSION

This paper presents a complete analysis of a simple stochastic zero-sum game. The information structure has the feature that each player knows his opponent's previous actions but not all of the data which determined those actions. Thus the information structure is not partially nested in the sense of [3]. Nonetheless, the game is sufficiently simple and structured that it is possible to solve the problem and characterize all saddle point strategies and, furthermore, to identify particular saddle point strategies which uniquely maximize the secondary criterion of penalizing an opponent for any suboptimal play.

REFERENCES

1. S. Karlin, *Mathematical Methods and Theory in Games, Programming, and Economics*, Vol. 2, Chapter 9, Reading: Addison-Wesley, 1959.

2. M. Loeve, *Probability Theory*, D. Van Nostrand Company, Inc., New York, 1955.

3. Y. C. Ho and K.-C. Chu, "Team Decision Theory and Information Structures in Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. AC-17, No. 1, pp. 15-28, 1972.

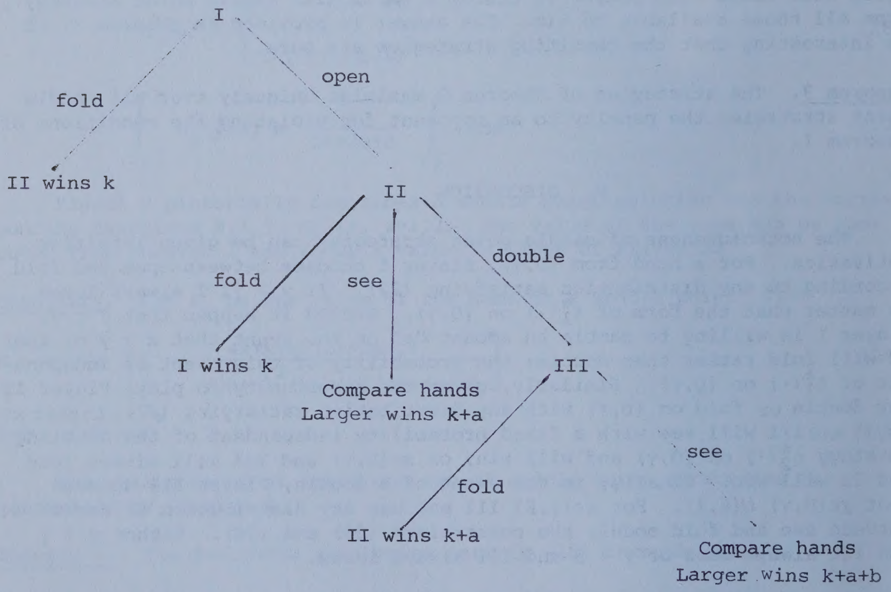

I
fold
open
II wins k
II
fold
see
double
I wins k
Compare hands
Larger wins k+a
III
see
fold
II wins k+a
Compare hands
Larger wins k+a+b

FIGURE 1 Game tree for simplified two-player poker

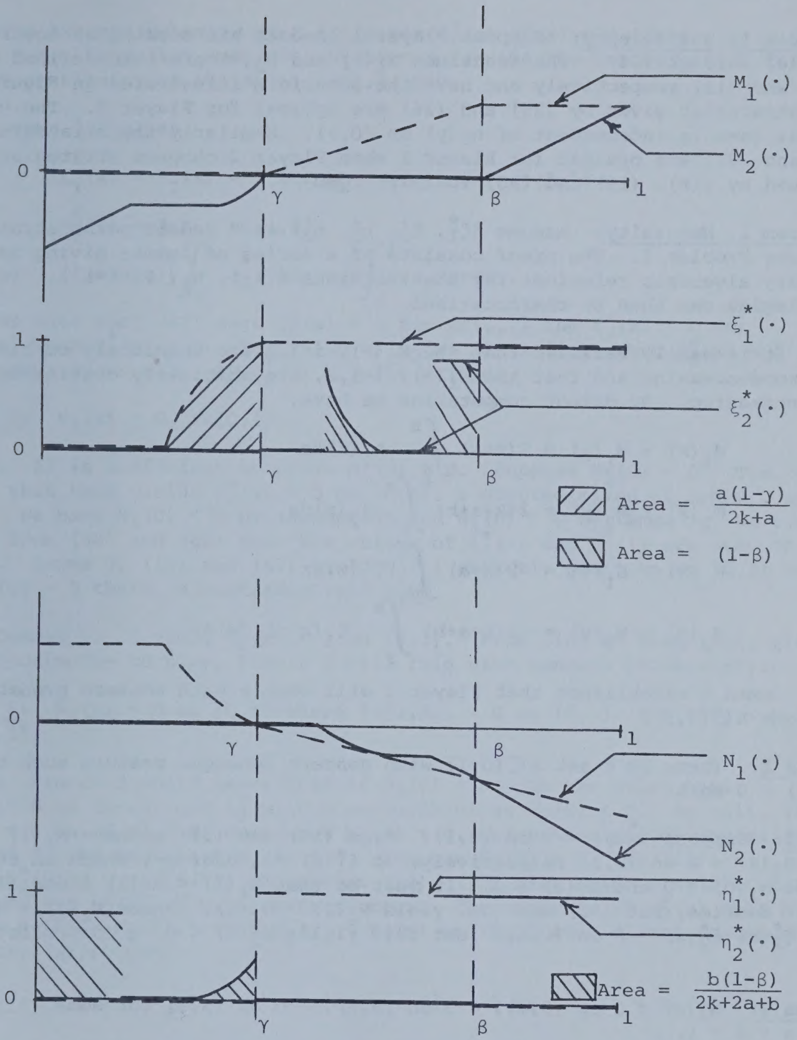


FIGURE 2 Saddle point strategies and the corresponding functions $M_i(\cdot)$ and $N_i(\cdot)$, $i = 1,2$.

APPENDIX

PROOFS OF THE THEOREMS

Theorem 1, Sufficiency: Suppose Player 1 chooses his strategies according to (18) through (24). The functions $N_1(\cdot)$ and $N_2(\cdot)$ are then defined by (12) and (13) respectively and have the same form illustrated in Figure 2. The strategies given by (25) and (26) are optimal for Player 2. The value of the game is independent of $\eta_2(y)$ on $[0,\gamma)$. Similarly the strategies (20) and (21) are optimal for Player 1 when Player 2 chooses strategies defined by (18), (19) and (25) to (27). QED

Theorem 1, Necessity: Assume $(\xi_1^*, \xi_2^*, \eta_1^*, \eta_2^*)$ is a saddle point strategy set for Problem 1. The proof consists of a series of lemmas giving necessary algebraic relations for the functions $M_i(\cdot)$, $N_i(\cdot)$, i=1,2. The strategies can then be characterized.

It is easily verified that the $M_i(\cdot)$, i=1,2 are absolutely continuous and nondecreasing and that the $N_i(\cdot)$, i=1,2, are absolutely continuous and nonincreasing. By direct computation we have:

$$M_1(x) - M_1(y) = 2(k+a)\int_y^x \eta_1(z)dz \tag{38}$$

$$M_2(x) - M_2(y) = 2(k+a+b)\int_y^x \eta_2(z)dz \tag{39}$$

$$N_1(x) - N_1(y) = -2(k+a)\int_y^x \xi_1(z)dz \tag{40}$$

$$N_2(x) - N_2(y) = -2(k+a+b)\int_y^x \xi_1(z)\cdot\xi_2(z)dz \tag{41}$$

Lemma 1 establishes that Player 2 will double with nonzero probability on some $A \subset [0,1]$.

Lemma 1: There is a set $A \subset [0,1]$ with nonzero Lebesque measure such that $\eta_2^*(y) > 0$ on A.

Proof: Suppose $\eta_2^*(y) = 0$ on $[0,1]$. From (10) and (11) we have $M_1(1) > 0$ and $M_2(x) = 0$ on $[0,1]$ respectively; so $\xi_1^*(x) = 1$ near x=1 which in turn gives $N_1(0) > 0$ and $N_1(1) < 0$. It must be that $N_1(1) \leq N_2(1)$ since II never doubles, but (12) and (13) yield $N_1(1) \geq N_2(1)$. Hence $N_1(1) = N_2(1)$ and $\xi_1^*(x)\cdot\xi_2^*(x) = 0$ on $[0,1]$. But this yields $N_2(0) < 0$, a contradiction. QED

Lemma 2: $M_2(x) < 0$ on $[0,\alpha)$, $= 0$ on $[\alpha,\beta]$, > 0 on $(\beta,1]$ for some $0 < \alpha < \beta < 1$.

Proof: From Lemma 1 and (11), $M_2(0) < 0$ and $M_2(1) > 0$. $M_2(x)$ is zero at most on an interval $[\alpha,\beta]$ where $0 < \alpha \leq \beta < 1$. The fact that $\alpha < \beta$ depends on Lemma 3.

Lemma 3: $\xi_1^*(x) = 1$ near $x = 1$.

Proof: By definition $\beta = \max\{x \mid M_2(x)=0,\ x\varepsilon[0,1]\}$ and we have $\beta < 1$. Since $M_2(x) > 0$ on $(\beta,1]$, $\xi_2^*(x) = 1$ there; so $\xi_1^*(x)$ on $(\beta,1]$ is determined by $M_1(x) + M_2(x)$ there. Algebra yields $M_1(1) + M_2(1) > 0$ and so $\xi_1^*(x) = 1$ near $x = 1$. QED

Completion of Proof to Lemma 2: Assume $M_2(x) = 0$ at a single point α. The proof establishes a $\delta > 0$ such that $N_1(y) < N_2(y)$ on $[\alpha,\alpha+\delta)$ and so $\eta_2^*(y) = 0$ there. But then by (39) $M_2(x) = 0$ on $[\alpha,\alpha+\delta)$, a contradiction.

The assumed behavior of $M_2(\cdot)$ gives $\xi_1^*(x)\cdot\xi_2^*(x) = 0$ for $x < \alpha$, $= \xi_1^*(x)$ for $x > \alpha$. From (12) and (13), $N_1(\alpha) < N_2(\alpha)$ and the contradiction follows by the continuity of the $N_i(\cdot)$, $i=1,2$. QED

Lemma 2 implies:

$$\xi_1^*(x) \cdot \xi_2^*(x) = 0 \text{ for } x\varepsilon[0,\alpha), = 1 \text{ for } x\varepsilon(\beta,1] \tag{42}$$

$$\eta_2^*(y) = 0, \qquad y\varepsilon(\alpha,\beta) \tag{43}$$

$$(2k+2a+b)\int_0^\alpha \eta_2^*(y)\,dy = b\int_\beta^1 \eta_2^*(y)\,dy\ . \tag{44}$$

We note that (42) says $\xi_2^*(x) = 1$ for $x\varepsilon(\beta,1]$ and $\xi_2^*(x) = 0$ for $x\varepsilon[0,\alpha)$ when $\xi_1^*(x) > 0$. The value of $\xi_2^*(x)$ for $x\varepsilon[0,\alpha)$ when $\xi_1^*(x) = 0$ is arbitrary; so we will say $\xi_2^*(x) = 0$ for $x\varepsilon[0,\alpha)$.

Lemma 4: $M_1(x) \geq 0$, $x\varepsilon[0,1]$.

Proof: It is sufficient to prove $M_1(0) \geq 0$. Suppose $M_1(0) < 0$. The proof shows that this yields $\eta_2^*(y) = 0$ on $[0,\alpha)$, a contradiction of Lemma 2 and (39). We have $M_1(0) < 0$ by assumption and $N_1(0) > 0$ by Lemma 3. We conclude from (38) and (40) that the values of $\xi_1^*(\cdot)$ and $\eta_1^*(\cdot)$ are zero on $[0,\beta)$. Lemma 3, (13) and (41) yield $N_2(\cdot)$ strictly positive on $[0,\beta)$ and so $\eta_2^*(y) = 0$ there, a contradiction. QED

Lemmas 2, 4 yield $\xi_1^*(x) = 1$ on $(\beta,1]$. From (10) we find that, given the opportunity to play, Player 2 will fold with nonzero probability.

Lemma 5: $N_2(y) = 0$ on $[0,\delta]$ where $\delta\varepsilon[\alpha,\beta)$, < 0 on $(\delta,1]$, $= N_1(\beta)$, $< N_1(y)$ on $(\beta,1]$.

Proof: Player 2 would never fold if $N_2(0) < 0$. On the other hand, $\eta_2^*(y) > 0$ on $(\alpha-,\alpha)$ and $\eta_2^*(y) = 0$ on (α,β) gives $N_2(\alpha) \leq 0$. By (41), (42), $N_2(y) = 0$ on $[0,\alpha)$. We next show $N_2(\beta) < 0$, which then yields the existence of δ. We know that $N_2(\beta) \leq 0$. Suppose $N_2(\beta) = 0$. By (41), $\xi_1^*(x)\cdot\xi_2^*(x) = 0$ on $[0,\beta)$. Equations (12), (13) give $N_1(\beta) < N_2(\beta)$. But this implies $\eta_2^*(\beta+) = 0$, a contradiction. The behavior of $\eta_2^*(y)$ near β yields $N_1(\beta) = N_2(\beta)$. That $N_2(y) < N_1(y)$ on $(\beta,1]$ is a direct consequence of (40), (41). QED

Lemma 6: $N_1(y) \geq 0$ on $[0,\gamma)$ where $\gamma\varepsilon[\alpha,\delta]$, < 0 on $(\gamma,1]$, $\leq N_2(y)$ on (γ,β).

Proof: The fact that $N_2(y) < 0$ on $(\delta,\beta]$ but $\eta_2^*(y) = 0$ there yields $N_1(y) \leq N_2(y)$ on $(\delta,\beta]$. The behavior of $\eta_2^*(y)$ near α gives $N_1(\alpha) \geq N_2(\alpha) = 0$. The existence of γ follows by the continuity of $N_1(\cdot)$. QED

Lemmas 5,6 and (43) imply conditions (27), (26) and part of (25). We now show $\eta_1^*(y) = 0$ on $[0,\gamma)$. Since $N_1(\cdot)$ is continuous and nonincreasing, $N_1(y) = 0$ either at a point γ or on an interval $[\mu,\gamma]$. If it is a point, we are finished. If it is an interval, $\xi_1^*(x) = 0$ on (μ,γ) by (40). But then $M_1(x) = 0$ on $[0,\gamma)$ and by (38) $\eta_1^*(y) = 0$ on $[0,\gamma]$.

Since Player 2 always sees on (γ,β) and never doubles, $N_1(y) \leq N_2(y)$; the inequality (24) results. Equations (38) and (25) yield $M_1(x) \geq 0$ on

$(\gamma,1]$ and so $\xi_1^*(x) = 1$ there. That $N_2(y) = 0$ on $[0,\gamma]$ gives $\xi_2^*(x) = 0$ there. Equation (22) results from $N_1(\gamma) = 0$ and (12), (23) from $N_1(\beta) = N_2(\beta)$ and (19) from $N_2(0) = 0$ and (13).

Lemma 7: $M_1(x) = 0$ on $[0,\gamma)$.

Proof: It suffices to show $M_1(0) = 0$. Lemma 4 gives $M_1(0) \geq 0$. Suppose $M_1(0) > 0$. Then $\xi_1^*(x) = 1$ on $[0,1]$. We find by computation that $M_1(0) < 0$, a contradiction.

Equation (18) follows from Lemma 7 and this complete the proof of Theorem 1. QED

Corollary Proof: Use equations (18) through (27) and compute K^* from (8) or (9).

Proof of Theorem 2: Algebra shows that (18), (19), (30), (31), (32) yield the parameters β, γ, ε, μ, δ such that $0 < \mu < \gamma$ and $0 < \delta < \gamma < \varepsilon < \beta < 1$. Except possibly for (24), it is obvious that the strategies of Theorem 2 satisfy all conditions of Theorem 1. It is clear that (24) is satisfied for $x\varepsilon[\varepsilon,\beta)$. We must verify that (24) is also satisfied for $x\varepsilon(\gamma,\varepsilon)$. This follows by noting that for any $x\varepsilon(\gamma,\varepsilon)$

$$\frac{k+a}{k+a+b}\int_x^\beta dz < \frac{k+a}{k+a+b}\int_\gamma^\beta dz = \int_\beta^1 dz = \int_\varepsilon^\beta dz . \quad \text{QED}$$

Proof of Theorem 3: The flexibility in $\xi_2^*(\cdot)$ only affects $N_2(y)$ on (γ,β). Suppose Player II uses the suboptimal strategies $\eta_1(y) = 1-g(y)$, $\eta_2(y) = h(y)$ where $1 \geq g(y) \geq h(y) \geq 0$ on (γ,β). The penalty to Player II for suboptimal play is $\int_\gamma^\beta [h(y)N_2(y) - g(y)N_1(y)]dy$. To maximize this penalty, Player III will strive to keep $N_2(y)$ as large as possible over (γ,β). When Players I, III play optimally, $N_2(y) = -2(k+a+b)\int_\gamma^y \xi_2^*(x)dx$ on (γ,β). For any feasible $g(\cdot)$ and $h(\cdot)$, the obvious choice is to set $\xi_2^*(x) = 1$ on the right hand portion of (γ,β) and zero on the left portion. Theorem 2 shows this is a saddle point strategy.

A similar analysis applies to the specification of a particular saddle point doubling strategy for Player II for $y\varepsilon[0,\gamma)$ and a particular saddle point opening strategy for Player I for $x\varepsilon[0,\gamma)$. QED

FREQUENCY-DOMAIN DESIGN OF FEEDBACK SYSTEMS FOR SPECIFIED INSENSITIVITY OF TIME-DOMAIN RESPONSE TO PARAMETER-VARIATION

K. R. KRISHNAN* and A. CRUICKSHANKS†
*Clarkson College of Technology, Potsdam, New York 13676
†Raytheon Company, Bedford, Massachusetts 01730

ABSTRACT

For the design of a linear, time-invariant minimum-phase feedback system to achieve specified insensitivity of response to parameter-variation, a method of translating response-specifications from time-domain to frequency-domain is introduced. This allows frequency-domain design methods to be used to achieve the specified insensitivity of time-domain response.

INTRODUCTION

Designing a feedback system so that its response to an input remains within specified bounds for given tolerances on parameters of the system is a problem of practical importance in automatic control.

Suppose $P(s,\alpha)$ is a linear, time-invariant minimum-phase plant with a set of parameters α which may assume values in a given closed and bounded set Ω. Then, by a design of the form shown in Fig. 1, it is possible to achieve the desired insensitivity of response to parameter-variation with sufficiently large bandwidth in the closed-loop system[1]. However, a large bandwidth tends to cause noise from the feedback sensor to be amplified at the plant-input and thus to drive the plant into saturation. An optimum design[1] may therefore be defined as one that satisfies the sensitivity specifications with the minimum transmission of sensor noise $n(t)$ to the plant.

In Fig. 1, let $L(s,\alpha) \triangleq P(s,\alpha)G(s)$.

Then $$T(s,\alpha) \triangleq \frac{C(s,\alpha)}{R(s)} = F(s)\,\frac{L(s,\alpha)}{1+L(s,\alpha)}.$$

A frequency-domain method of designing such a feedback system has been developed by Horowitz and Sidi[1]. The bounds on the response, if given in the time-domain, are assumed to be first translated into bounds in the frequency-domain on the transfer-function $T(j\omega,\alpha)$. These frequency-domain bounds are then used to derive certain loci on the Nichols Chart for the loop transfer-function $L(j\omega,\alpha_o)$ at some nominal plant-condition α_o. These loci define admissible values of $L(j\omega,\alpha_o)$ at different frequencies and enable the optimum design to be determined.

However, in Horowitz and Sidi's work, the initial translation of time-domain specifications on the response into corresponding frequency-domain bounds is left largely to trial-and-error methods. We propose here a rational method of converting time-domain specifications to frequency-domain bounds. Once the frequency-domain bounds are obtained, the design can proceed as in Horowitz and Sidi[1]. However, for a special choice of the pre-filter $F(s)$, a simple sub-optimal design is possible for which the Nichols Chart loci are particularly simple to derive and interpret. The sub-optimal design is attractive because of the simple derivation of the design loci.

TRANSLATION OF TIME-DOMAIN SPECIFICATIONS INTO FREQUENCY-DOMAIN CONDITIONS

In the system of Fig. 1, $P(s,\alpha)$ is a linear, time-invariant minimum-phase plant with parameters α which can assume values in a closed and bounded set Ω. Suppose $F(s)$ and $G(s)$ are to be designed such that the response $c(t,\alpha)$ to a command-input $r(t)$ lies within the bounds

$$a(t) \leq c(t,\alpha) \leq b(t), \quad t \geq 0, \text{ for all } \alpha\varepsilon\Omega. \tag{1}$$

$$\text{Let } m(t) \triangleq \frac{b(t)+a(t)}{2}, \text{ and } v(t) \triangleq \frac{b(t)-a(t)}{2}. \tag{2}$$

Using (2), (1) may be replaced by the equivalent condition

$$[c(t,\alpha)-m(t)]^2 \leq v^2(t), \quad t \geq 0, \ \alpha\varepsilon\Omega \tag{3}$$

Condition (3), as it stands, is difficult to convert to the frequency-domain. It is suggested that a reasonable alternative to (3) is the slightly weaker condition

$$\int_0^t [c(p,\alpha)-m(p)]^2 dp \leq \int_0^t v^2(p)dp, \quad t \geq 0, \ \alpha\varepsilon\Omega \tag{4}$$

It is assumed that the bounds $a(t)$ and $b(t)$ are such that each integral converges as $t\to\infty$.

A sufficient condition in the frequency-domain corresponding to the time-domain condition (4) is given by [2,3]

$$\underset{\alpha\varepsilon\Omega}{\text{Max}} \left| \frac{C(j\omega,\alpha)}{M(j\omega)} - 1 \right| \leq \left| \frac{V(j\omega)}{M(j\omega)} \right|, \text{ for all } \omega. \tag{5}$$

Substituting $C(s,\alpha) = R(s)T(s,\alpha)$, we have

$$\underset{\alpha\varepsilon\Omega}{\text{Max}} \left| \frac{R(j\omega)\ T(j\omega,\alpha)}{M(j\omega)} - 1 \right| \leq \left| \frac{V(j\omega)}{M(j\omega)} \right|, \text{ for all } \omega. \tag{6}$$

Condition (6) is a frequency-domain statement corresponding to the parameter-insensitivity of response specified in the time-domain by (4).

At each ω, (6) imposes constraints on both the magnitude and phase of $T(j\omega,\alpha)$. In a minimum-phase transfer-function, the magnitude function determines the phase function as well; however, it is not clear that bounds on the magnitude determine corresponding bounds on the phase. Such an assumption, it may be noted, would enable (6) to be replaced by the corresponding conditions on $|T(j\omega,\alpha)|$:

$$\text{Max}\ \{0, \frac{|M(j\omega)|-|V(j\omega)|}{|R(j\omega)|}\} \leq |T(j\omega,\alpha)| \leq \frac{|M(j\omega)|+|V(j\omega)|}{|R(j\omega)|} \quad \text{for } \alpha\varepsilon\Omega \text{ and all } \omega. \tag{7}$$

Condition (7) is of the form which is the starting point of design by Horowitz and Sidi's method.

In this paper, however, the above assumption is not made. Instead, a simple sub-optimal design is obtained directly from (6).

A SIMPLE SUB-OPTIMAL DESIGN

From Fig. 1, $$T(s,\alpha) = F(s)\,\frac{L(s,\alpha)}{1+L(s,\alpha)}, \tag{8}$$

where $$L(s,\alpha) \triangleq P(s,\alpha)G(s) \tag{9}$$

If the pre-filter F(s) is chosen to be

$$F(s) = \frac{M(s)}{R(s)}\ , \tag{10}$$

we have, on substituting for $T(j\omega,\alpha)$ in (6) from (8),

$$\operatorname*{Max}_{\alpha\varepsilon\Omega} |S(j\omega,\alpha)| \le \left|\frac{V(j\omega)}{M(j\omega)}\right| \triangleq A(\omega), \tag{11}$$

where $S(s,\alpha) \triangleq \dfrac{1}{1+L(s,\alpha)}$, the logarithmic sensitivity[4] of $T(s,\alpha)$ with respect to $L(s,\alpha)$ or $P(s,\alpha)$.

With F(s) chosen by (10) for convenience, a design of G(s) to secure the specified insensitivity of response with minimum effect of sensor-noise is, of course, a sub-optimal design for the original problem.

In addition to specifications on response to command-input, some restriction is also placed, as a rule, on the response to an output disturbance (d(t) in Fig. 1), which is governed by the transfer-function $\dfrac{C(s,\alpha)}{D(s)} = \dfrac{1}{1+L(s,\alpha)} = S(s,\alpha)$. It is usual to require a certain minimum amount of damping in the disturbance-response, expressed by a condition of the form

$$\operatorname*{Max}_{\alpha\varepsilon\Omega}\ \operatorname*{Max}_{\omega} |S(j\omega,\alpha)| \le \beta\ . \tag{12}$$

Combining (11) and (12), we arrive at the design criterion

$$\operatorname*{Max}_{\alpha\varepsilon\Omega} |S(j\omega,\alpha)| \le \text{Min}\ \{A(\omega),\beta\}. \tag{13}$$

INTERPRETATION OF THE DESIGN CRITERION ON THE NICHOLS CHART[1,5]

The Nichols Chart provides a simple interpretation of (13).

Write $$S(s,\alpha) = \frac{1}{1+L(s,\alpha)} = \frac{Q(s,\alpha)}{1+Q(s,\alpha)}\ , \tag{14}$$

$$\text{where}\quad Q(s,\alpha) \triangleq \frac{1}{L(s,\alpha)}\ . \tag{15}$$

Then (13) may be written as

$$\operatorname*{Max}_{\alpha\varepsilon\Omega} \left|\frac{Q(j\omega,\alpha)}{1+Q(j\omega,\alpha)}\right| \le \text{Min}\ \{A(\omega),\ \beta\},\ \text{for all}\ \omega \tag{16}$$

Let α_o be some selected value of α in its range.

Then $$\frac{Q(s,\alpha)}{Q(s,\alpha_o)} = \frac{L(s,\alpha_o)}{L(s,\alpha)} = \frac{P(s,\alpha_o)G(s)}{P(s,\alpha)G(s)} = \frac{P(s,\alpha_o)}{P(s,\alpha)}$$

$$\left.\begin{aligned} &\text{Therefore,} \quad |Q(j\omega,\alpha)|_{db} = |Q(j\omega,\alpha_o)|_{db} + \left|\frac{P(j\omega,\alpha_o)}{P(j\omega,\alpha)}\right|_{db}, \\ &\text{and} \quad \operatorname{Arg} Q(j\omega,\alpha) = \operatorname{Arg} Q(j\omega,\alpha_o) + \operatorname{Arg}\left[\frac{P(j\omega,\alpha_o)}{P(j\omega,\alpha)}\right]. \end{aligned}\right\} \quad (17)$$

The range of parameter-variation determines the range of possible values of $\left|\frac{P(j\omega,\alpha_o)}{P(j\omega,\alpha)}\right|_{db}$ and Arg $\left[\frac{P(j\omega,\alpha_o)}{P(j\omega,\alpha)}\right]$, for each ω. Hence, for each ω, for any assumed value of $Q(j\omega,\alpha_o)$, the possible range of values of $|Q(j\omega,\alpha)|_{db}$ and Arg $Q(j\omega,\alpha)$ are determined by (17). By locating the assumed value of $Q(j\omega,\alpha_o)$ on the Nichols Chart and considering the range of possible values of $Q(j\omega,\alpha)$, the corresponding range of variation of $\frac{Q(j\omega,\alpha)}{1+Q(j\omega,\alpha)}$ is directly read on the Chart.

Thus the Chart may be used to determine, for each assumed value of Arg $Q(j\omega,\alpha_o)$, the corresponding bounds on the value of $|Q(j\omega,\alpha_o)|_{db}$ which ensure that (16) is satisfied. When this is repeated for different assumed values of Arg $Q(j\omega,\alpha_o)$, a locus is generated on the Nichols Chart which forms the boundary of the admissible region for $Q(j\omega,\alpha_o)$. Since $|L(j\omega,\alpha_o)|_{db} = -|Q(j\omega,\alpha_o)|_{db}$ and Arg $L(j\omega,\alpha_o) = -$ Arg $Q(j\omega,\alpha_o)$, such a locus for $Q(j\omega,\alpha_o)$ can be immediately translated into a corresponding locus $B(\omega)$ for $L(j\omega,\alpha_o)$. Thus, for each ω, (16) determines a locus $B(\omega)$ which defines the admissible values of $L(j\omega,\alpha_o)$. The construction of these loci will become clear in the example given later.

It can be shown, using a theorem due to Horowitz[6], that for a given excess e of poles over zeros in $L(s,\alpha_o)$, the loci obtained above determine a unique minimum-phase $L(s,\alpha_o)$. At each ω, $L(j\omega,\alpha_o)$ lies on the corresponding locus $B(\omega)$, and is optimum in the sense that it satisfies (16) with the smallest value of $\lim_{s\to\infty} s^e L(s,\alpha_o)$.

In practice, of course, the design is based on just a few loci, constructed for a few frequencies in the frequency-range that is significant for the particular problem. An $L(s,\alpha_o)$ is then found, for a specified excess of poles over zeros, which lies close to the corresponding loci at these frequencies.

AN EXAMPLE OF DESIGN

In the system of Fig. 1, F(s) and G(s) are to be designed for the plant $P(s,k) = \frac{k}{s}$, $1 \leq k \leq 10$, such that the step-response c(t,k) lies between the function $a(t) = 1-e^{-t}(1+2t)$ and the function $b(t) = 1-e^{-t}$, $t \geq 0$ for $1 \leq k \leq 10$, at least in the sense of (4). The loop-transfer-function $L(s,k) = P(s,k)G(s)$ is to have an excess of 4 poles over zeros.

In addition, as a constraint on the disturbance-response, the condition

$$\max_{1 \leq k \leq 10} \max_{\omega} \left| \frac{1}{1+L(j\omega,k)} \right| \leq 6 \text{ db} \tag{18}$$

is also to be satisfied. On a second-order model, this would correspond to an overshoot of about 45% in the disturbance-response.

We have $r(t) = 1$, $m(t) = 1-e^{-t}(1+t)$, and $v(t) = te^{-t}$, $t \geq 0$. Hence, by (10),

$$F(s) = \frac{M(s)}{R(s)} = \frac{1}{(s+1)^2} \tag{19},$$

and

$$A(\omega) = \left| \frac{V(j\omega)}{M(j\omega)} \right| = \omega. \tag{20}$$

With $Q(s,k) = \frac{1}{L(s,k)}$, the design criterion is, from (16),

$$\max_{1 \leq k \leq 10} \left| \frac{Q(j\omega,k)}{1+Q(j\omega,k)} \right|_{db} \leq \text{Min } \{20 \log \omega,\ 6\}. \tag{21}$$

Let $k_o = 1$. Then $\frac{P(s,k_o)}{P(s,k)} = \frac{1}{k}$ and hence, by (17),

$$\left.\begin{aligned} |Q(j\omega,k)|_{db} &= |Q(j\omega,1)|_{db} - 20 \log k, \\ \text{and} \quad \text{Arg } Q(j\omega,k) &= \text{Arg } Q(j\omega,1) \end{aligned}\right\}, \quad 1 \leq k \leq 10 \tag{22}$$

Fig. 2 shows the Nichols Chart loci $B(\omega)$ derived for $L(j\omega,1)$ from (21) and (22), for the frequencies $\omega = 0.1, 0.2, 0.4, 0.7, 1$ and 2 rad/sec. The locus $B(1)$ at $\omega = 1$, for example, is obtained as follows: we have, from (21),

$$\max_{1 \leq k \leq 10} \left| \frac{Q(j,k)}{1+Q(j,k)} \right|_{db} \leq 0 \text{ db}.$$

It follows from (22) that $|Q(j,k)|_{db}$ decreases by 20 db as k goes from 1 to 10 while Arg $Q(j,k)$ does not change. Examining the contours of constant magnitude of closed-loop function on the Nichols Chart, it follows that if, for example, Arg $Q(j,1) = 100°$, then $|Q(j,1)|_{db}$ can be no larger than about 8 db. Note that the contours of constant closed-loop magnitude are symmetric about the 0° ordinate and hence the -100° ordinate may be used instead of the 100° ordinate for this purpose. Thus, if Arg $L(j,1) = -100°$, then $|L(j,1)|_{db}$ can be no smaller than -8 db, the point marked J on Fig. 2. Repeating this for other assumed values of Arg $Q(j,1)$, the locus $B(1)$ is obtained as the lower boundary of the admissible region for $L(j,1)$. For this problem, the loci $B(\omega)$ are simple contours for $\omega \leq 1$ and closed contours for $\omega > 1$. Condition (18) on the disturbance-response gives rise to the closed-contour marked D on Fig. 2, which is the same as the locus $B(2)$. For $\omega > 2$, the loci $B(\omega)$ are contained within D. Hence, in view of (21), the design of $L(j\omega,1)$ is based on the loci $B(\omega)$ for $\omega \leq 2$ and on the closed contour D for $\omega > 2$.

The minimum-phase function $L(s,1) = \dfrac{1234.8(s+4)(s+5.5)}{s(s+2)(s+2.2)(s+25)(s^2+5.6s+196)}$ has the required excess of 4 poles over zeros and meets the design requirements fairly closely, as shown by the graph of $L(j\omega,1)$ in Fig. 2. Since $L(s,1) = P(s,1)G(s) = \dfrac{G(s)}{s}$, we have

$$G(s) = \frac{1234.8(s+4)(s+5.5)}{(s+2)(s+2.2)(s+25)(s^2+5.6s+196)}, \tag{23}$$

and the design is complete.

The step-responses of the system for several values of k are shown in Fig. 3, together with the specified bounds. Of course, as seen from (4), the system was designed to meet the specifications in the integral-square sense, but Fig. 3 shows that, in fact, the specifications are very closely met in the stronger sense of (1).

Fig. 4 shows the responses of the system for several values of k to a step-disturbance at the output. The overshoot in the transient oscillations does not exceed about 48%, a figure that corresponds to the disturbance-response constraint given in (18).

CONCLUSION

For the design of a linear, time-invariant, minimum-phase feedback system to realize a specified insensitivity of response to parameter variation, a method has been proposed to convert time-domain response specifications into frequency-domain conditions. These frequency-domain conditions enable the system to be designed[1], in the configuration of Fig. 1, to meet the sensitivity specifications while minimizing the effect of feedback sensor noise on the plant. A special (sub-optimal) choice of the pre-filter F(s) leads to a very simple design criterion on the loop-transfer-function (or the logarithmic sensitivity function) which is readily interpreted and exploited for design on the Nichols Chart. The method is illustrated by means of an example.

REFERENCES

1. I. M. Horowitz and M. Sidi, "Synthesis of Feedback Systems with Large Plant-Ignorance for Prescribed Time-Domain Tolerances," Int. J. of Cont., Vol. 16, No. 2, pp. 287-309, 1972.

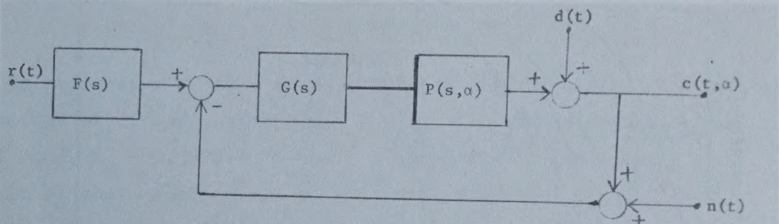
2. J. B. Cruz, Jr. and W. R. Perkins, "A New Approach to the Sensitivity Problem in Multivariable Feedback System Design," IEEE Trans. Auto. Cont., Vol. AC-9, No. 3, pp. 216-223, 1964.

3. E. Kreindler, "On the Definition and Application of the Sensitivity Function," J. of Franklin Inst., Vol. 285, No. 1, pp. 26-36, 1968.

4. I. M. Horowitz, Synthesis of Feedback Systems, Academic Press, 1963, Chapter 3.

5. A. Gelb, "Graphical Evaluation of the Sensitivity Function Using the Nichols Chart," IRE Trans. Auto. Cont., Vol. AC-7, No. 4, pp. 57-58, 1962.

6. I. Horowitz, "Optimum Loop-Transfer Function in Single Loop Minimum Phase Feedback Systems," Int. J. of Cont., Vol. 18, No. 1, pp. 97-113, 1973.



$$L(s,\alpha) \triangleq P(s,\alpha)G(s)$$

$$T(s,\alpha) \triangleq \frac{C(s,\alpha)}{R(s)} = F(s)\,\frac{L(s,\alpha)}{1+L(s,\alpha)}$$

Fig. 1. Configuration for System Design

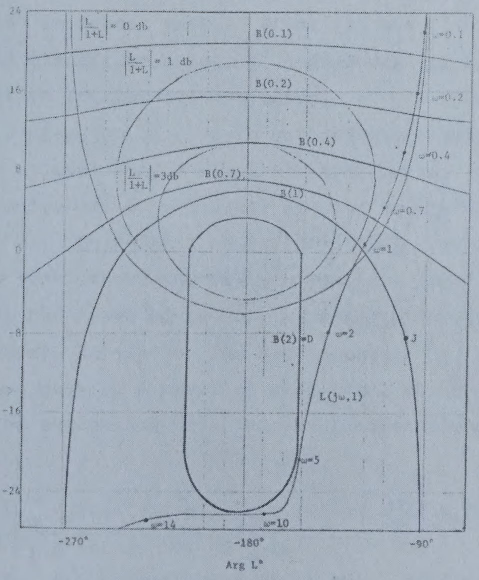


Fig. 2. Design Loci B(ω) for L(jω,1) on the Nichols Chart.

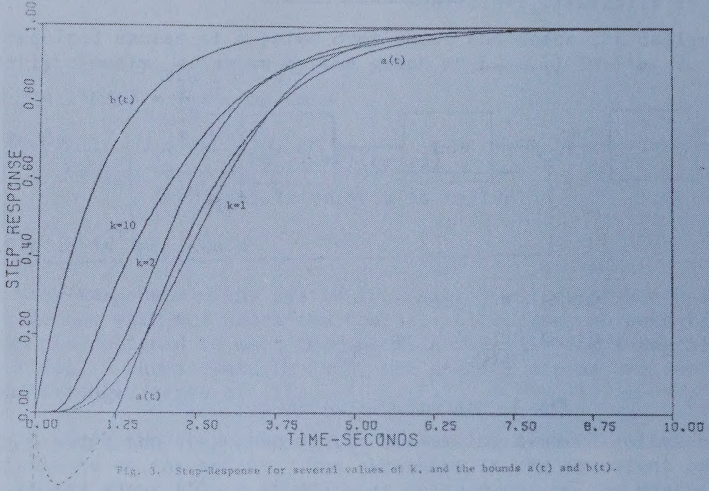


Fig. 3. Step-Response for several values of k, and the bounds a(t) and b(t).

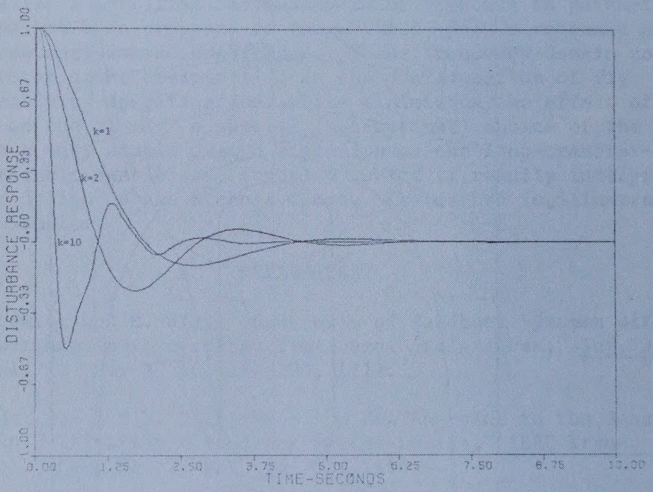


Fig. 4. Response to a step-disturbance at the output for several values of k.

NEAR OPTIMAL ALGORITHM FOR THE PARALLEL EVALUATION OF LINEAR RECURRENCES

AMNON BARAK and MARC SNIR

Department of Computer Science
The Hebrew University of Jerusalem
Jerusalem, Israel.

ABSTRACT

In this paper we develop near optimal algorithm for the parallel evaluation of linear recurrences x_n of the form $x_i = x_{i-1}A_i + B_i$, $i \geq 1$ and $x_0 = y$, where y is a given Horner expression.

It is shown that using p independent processors, a linear recurrence, or equivalently, a Horner expression can be evaluated in $3n/(2p+1) + O(p)$ time units and that this result is $O(p)$ close to the lower bound for the problem.

1. INTRODUCTION

Consider a p-processor parallel computer such as the ILLIAC IV. In this paper we develop an algorithm for the solution of first order linear recurrences which utilize the multiprocessing capability of such a computer. Because of the importance of the evaluation of arithmetic expressions, the problem of parallel evaluation of these expressions has been investigated in recent years. Brent [1] was the first to investigate parallel algorithms using p independent processors, where p is a positive integer independent of n. He showed that a general arithmetic expression E_n of n variables and without division can be evaluated in at most $2n/p + O(\log n)$ time units. Winograd [5] improved Brent's results and showed that E_n can be evaluated in at most $3n/2p + O(\log^2 n)$. He further conjectured that for $p > 1$, $3n/2p+$ lower term in n, is the lower bound for the problem. Hyafil and Kung [2] proved that the lower bound for the computation of linear recurrences is at least $3n/(2p+1)$ for all values of n and p.

In this paper we develop a parallel algorithm, which is easy for implemtation, for the evaluation of alternating expressions E_{2n} of $2n$ variables, denoted by

$$E_{2n} = (\ldots(a_0 \eta_1 b_1)\theta_1 a_1)\ldots\ )\eta_n b_n)\theta_n a_n,$$

where $\eta_i \in \{*\}$ and $\theta_i \in \{+,-\}$ for $i \geq 1$.

We show that E_n can be evaluated in at most $3n/(2p+1) + O(p)$ steps, by using p independent processors in cases which do not include divisions, thus achieving a speed up of $2p/3 + \frac{1}{3}$ over the serial computation with $p = 1$. If we compare the bound that we achieved to the proven lower bound in [2] we conclude that our result is asymptotically optimal for this case.

We note that in [4] we developed algorithms for the parallel evaluation of general alternating expression, polynomial expression and general rational functions of n variables in $5n/(2p+3) + O(p)$, $3n/(2p+1) + O(p^2)$ and $5n/(2p+3) + O(p^2)$ steps respectively.

2. ALTERNATING EXPRESSIONS

In this section we develop algorithms for the parallel evaluation of alternating expressions of $2n$ variables using p processors.

Let E_{2n} be an alternating expression,

$$E_{2n} = (\ldots(a_0 \eta_1 b_1)\theta_1 a_1)\ldots)\eta_n b_n)\theta_n a_n$$

where $\eta_i \in \{*\}$ and $\theta_i \in \{+,-\}$.

To compute E_{2n} we define a set of linear recurrences relations:

$$y_0 = a_0$$

$$(2.1) \qquad y_i = (y_{i-1}\eta_i b_i)\theta_i a_i, \qquad i = 2,3,\ldots,n,$$

where

$$E_{2n} = y_n$$

The recurrence relation (2.1) consists of n equations. Since only p processors are available we combine successive relations into groups of relations such that we have p recurrence relations.

Define

$$x_0 = (\ldots(a_0 \eta_1 b_1)\theta_1 a_1)\ldots)\eta_{k_1} b_{k_1})\theta_{k_1} a_{k_1}$$

$$x_1 = (\ldots(x_0 \eta_{k_1+1} b_{k_1+1})\theta_{k_1+1} a_{k_1+1})\ldots)\eta_{k_2} b_{k_2})\theta_{k_2} a_{k_2}$$

$$(2.2) \qquad = x_0 {}^* A_1 + B_1$$

$$x_2 = x_1 {}^* A_2 + B_2$$

$$x_{p-1} = x_{p-2} {}^* A_{p-1} + B_{p-1} = E_{2n}$$

We now introduce parallelism into the computation of (2.2) by computing the coefficients A_i, B_i of x_i, $i = 1,2,\ldots p-1$ simultaneously and independently in $p-1$ processors and, at the same time, we compute x_0. Our strategy is to choose $k_1, k_2, \ldots, k_p$ in such a way that when the computation of x_0 is done we also have A_1 and B_1. Therefore the time for the computation of x_1, is

$$t_1 = t_0 + 2.$$

Similarly we have

$$t_i = t_{i-1} + 2$$
$$= t_0 + (i-1)2, \ 1 \leq i \leq p-1.$$

The total time for all the computation is therefore

$$t_{p-1} = t_0 + (p-1)2.$$

The analysis that follows is intended for the selection of k_1, in x_0 and for the number of operation needed to compute A_i and B_i of x_i.

Lemma 2.1:

Let $x_n = (\dots(x_0 \eta_1 b_1)\theta_1 a_1)\dots)\eta_n b_n)\theta_n a_n$

where $\eta_i \in \{*\}$ and $\theta_i \in \{+,-\}$.

Then

$$x_n = x_0 * A_n \pm B_n$$

where A_n and B_n can be evaluated in $\max(3n-3,0)$ operations.

Proof:

For $i = 0,1$ the assertion is evident.

Suppose that

$$x_{n-1} = A_{n-1} * x_0 + B_{n-1}$$

where A_{n-1}, B_{n-1} can be evaluated in $3(n-1)-1$ operations. Then we can write for x_n

$$x_n = (A_{n-1} \eta_n b_n) * x_0 + (B_{n-1} \eta_n b_n)\theta_n a_n.$$

Define

$$A_n = A_{n-1} \eta_n b_n$$

and

$$B_n = (B_{n-1} \eta_n b_n)\theta_n a_n$$

and the conditions of the lemma are satisfied.

We now prove a theorem which establishes a bound for the number of steps required to evaluate an alternating expression without division.

Theorem 2.2:

Let $E_{2n} = (\dots(a_0 \eta_1 b_1)\theta_1 a_1)\dots)\eta_n b_n)\theta_n a_n$

where

$\eta_i \in \{*\}$ and $\Theta_i \in \{+,-\}$.

E_{2n} can be evaluated with p processors in $\frac{6n}{2p+1} + p + 2$ steps.

Proof:

We first assume the case

(2.3) $\quad n > \frac{1}{3}(p+2)(p-1)$.

Let $0 \leq k_1 \leq k_2 \leq \ldots \leq k_p = n$.

Define

$$x_i = (\ldots(a_0 \eta_1 b_1)\Theta_1 a_1)\ldots)\eta_{k_i} b_{k_i})\Theta_{k_i} a_{k_i}$$

for $i = 1,2\ldots,p$

where $x_p = E_{2n}$.

According to lemma 2.1

$$x_i = x_{i-1} * A_i \pm B_i, \quad i = 2,3,\ldots,p$$

where A_i and B_i can be computed in $\max(3P_i-3,0)$ operations where $P_i = k_i - k_{i-1}$ for $2 \leq i \leq p$.

We shall now use processor p_i for the computation of x_i, i.e. A_i and B_i.

Let T_i be the number of steps necessary to evaluate x_i, then it is clear that

$$T_1 \leq 2P_1 \quad \text{where} \quad P_1 = k_1.$$

For $i \geq 2$, x_{i-1} is already computed in $\max(T_{i-1}, 3P_i - 4)$, as is A_i, and we need at most one more step to complete the computation of B_i. Two more steps are required now for the computation of x_i. During the first step we compute $x_{i-1} * A_i$ in one processor and complete the computation of B_i in the other. During the second step we evaluate x_i.

We have

$$T_i \leq \max(T_{i-1}, 3P_i - 4) + 2, \quad 2 \leq i \leq p,$$

therefore

$$T_p \leq \max(2P_1 + 2(p-1), \quad 3P_i - 2 + 2(p-i)), 2 \leq i \leq p.$$

We will complete the proof of the theorem by showing a set of P_i which satisfies all the requirements.

Let

$$\tilde{P}_1 = \frac{3n-(p-1)(p+2)}{2p+1}$$

and

$$\tilde{P}_i = \frac{2}{3}(P_1 + i), \quad i = 2,3,\ldots,p.$$

Then we have from (2.3)

$$\tilde{P}_i > 0$$

and

$$\sum_{i=1}^{p} p_i = n.$$

We now have

$$2P_1 + 2(p-1) = 3\tilde{P}_i - 2 + 2(p-1), \quad 2 \leq i \leq p$$

so that

$$\tilde{T}p = \max_{2 \leq i \leq p} \left(2P_1 + 2(p-1), \quad 3P_i - 2 + 2(p-1)\right)$$

$$= 2\tilde{P}_1 + 2(p-1) = p - \frac{5}{2} + \frac{9}{4p+2} + \frac{6n}{2p+1}.$$

Choose P_i for $1 \leq i \leq p$ such that

$$|P_i - \tilde{P}_i| < 1,$$

$$P_i \geq 0$$

$$\sum_{i=1}^{p} P_i = n$$

and

$$k_i = \sum_{j=1}^{i} P_j.$$

then the algorithms can evaluate E_{2n} in T_p steps where

$$T_p \leq \tilde{T}_p + 3 \leq \frac{6n}{2p+1} + p + 1/2 + \frac{9}{4p+2}$$

or

$$T_p \leq \left\lfloor \frac{6n}{2p+1} \right\rfloor + p + 2$$

and the theorem is proved for $n > \frac{1}{3}(p+2)(p-1)$. It is clear that the theorem holds for $p = 1$ and for $n \leq p + 2$.

If $p > 1$ and $\bar{p} + 2 < n < \frac{1}{3}(p+2)(p-1)$ we choose q to be $q = \max \{r: n > \frac{1}{3}(r+2)(r-1)\}$.

We can compute E_{2n} using q processors in T_q steps where

$$T_q \leq \frac{6n}{2p+1} + q + 1/2 + \frac{9}{4q+2},$$

and we are still within the bounds of the theorem.

We note that it is evident that for small n better algorithms can be developed.

REFERENCES

[1] Brent, R.P. The parallel evaluation of arithmetic expressions in logarithmic time. Complexity of sequential and parallel numerical algorithms, J.F. Traub, Edition, Academic Press, 1973.

[2] Hyafil, L., and Kung, H.T. The complexity of parallel evaluation of linear recurrences. Proceedings of the seventh annual ACM Symponium on theory of Computing, Albuquerque, New Mexico, pp. 12-22, May 1975.

[3] Kuck, D.J., and Maruyama, K. The parallel evaluation of arithmetic expressions of special forms. IBM Thomas J. Watson Research Center, RC - 4276, March 1973.

[4] Snir, M., and Barak, A. On the parallel evaluation of rational expressions, Report No. 14, Department of Computer Science, The Hebrew University of Jerusalem, Jerusalem, Israel, January 1975.

[5] Winograd, S. On the parallel evaluation of certain arithmetic expressions. IBM Thomas J. Watson Research Center, RC - 4808, April 1974.

TWO METHODS FOR IMPROVING THROUGHPUT OF A PIPELINE*

WATARU MAYEDA
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

ABSTRACT

One of the two methods introduced here is to modify all types of tasks by the use of pseudo segments so that every task has a repeating form of one basic sequence. Because of repeating the same basic sequence, each task can be initiated whenever the first segment necessary for computing a waiting task is vacant without worry about collision with tasks already in a pipeline. Obtaining a suitable basic sequence which yields the maximum throughput is achieved by the use of a graph called a combined task graph.

Another method introduced here is to employ an additional buffer register at each segment so that when more than one task wish to be processed at the same segment at the same time, tasks which can not be processed right away can be transferred into these buffer registers.

By adding another buffer register associated with the buffer register at the output of each segment, one can hold at most two tasks at a time when the segment is not used for processing. Under this circumstance, a waiting task can be initiated whenever the first segment required to compute a waiting task, is not occupied by a task already in a pipeline. It will be shown that there is always a way of transferring tasks in a pipeline so that no collision will develop.

Finally, the throughput of these two methods and that of a method by scheduling initiation time for waiting tasks are compared.

1. INTRODUCTION

A pipeline is a collection of hardware segments which can be operated simultaneously. Each computation task requires a particular sequence of segments in time, which can be indicated by an ordered set $\{S_1\ S_2\ ---\}$ with the assumption that the time required to complete a task at each segment is the same. There is a method called an optimum scheduling of tasks so that no collision will occur in a pipeline [1] where a collision means to occupy a segment by more than one task at the same time. Here, we will introduce two new methods one of which is to modify all tasks so that once tasks are in a pipeline, no collision will occur. Hence, whenever there is a segment which is not occupied by a task in a pipeline and a waiting task has that segment as the first segment of its sequence, the task can be initiated.

The second method is to employ an additional buffer register for each segment in a pipeline so that when a task can not be transferred to a desired segment, it can be stored in a buffer register until the segment becomes vacant.

---

*This work was supported by the Joint Services Electronics Program (U.S. Army, U.S. Navy, U.S. Air Force) under Contract DAAB-07-72-C-0259.

Finally, the throughputs under these two methods and a method of optimum scheduling of tasks [1] will be compared.

2. MODIFIED TASKS

We will develop a procedure of modifying given tasks to a repeating form of one basic sequence so that whenever the first segment of a modified task is vacant in a pipeline, we can initiate the task.

2.1. Efficiency Index

Consider a task $A = \{S_1S_2S_3S_1S_3\}$. Table 1 shows a necessary segment at each unit of time. Such a table is called a reservation table.

Table 1. Reservation Table

$$\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
S_1 & A & & & A & \\
S_2 & & A & & & \\
S_3 & & & A & & A \\
\hline
\end{array}$$

Suppose task A_1 is initiated at $t = 1$ and task A_2 is initiated at $t = 2$. To show which tasks are in which segment at each time, we use a flow chart as shown in Table 2 where ☐ indicates the last computation for each task. Note that another task A_3 can not be initiated until $t = 6$. One method of obtaining an initiation time for a next task without producing any collision in a pipeline is a method of optimum scheduling [1].

Table 2. Initiation of Two Tasks A_1 and A_2

$$\begin{array}{c|cccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
S_1 & A_1 & A_2 & & A_1 & A_2 & \\
S_2 & & A_1 & A_2 & & & \\
S_3 & & & A_1 & A_2 & \boxed{A_1} & \boxed{A_2} \\
\hline
\end{array}$$

Because the next task can not be initiated until $t = 6$, the average latency* is 2.5.

2.2. Pseudo Segment

Suppose we can use any segment not for computation but just holding data (which is equivalent to computing data in such a way as to make the data invariant.) A segment which is employed for just holding data is named a "pseudo segment" and is symbolized by encircling the segment. Consider a task $A = \{S_1S_2S_3S_1S_3\}$ in the previous section. Inserting a

---

*An average latency in the average time that a task must wait before it can be initiated.

pseudo segment S_2 as

$$A^1 = \{S_1S_2S_3S_1\textcircled{S_2}S_3\}$$

obviously makes the task invariant. However the sequence becomes a repeating form of a so called basic sequence $\{S_1S_2S_3\}$. It is clear that such a repeating form has an advantage which is whenever a task can be initiated, it will never collide with other tasks already in a pipeline. As an example, a flow chart when task A^1 is initiated consecutively is shown in Table 3.

Table 3. Consecutive Initiation of Task A^1

$$\begin{array}{c|cccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
S_1 & A_1 & A_2 & A_3 & A_1 & A_2 & A_3 & & \\
S_2 & & A_1 & A_2 & A_3 & \textcircled{A_1} & \textcircled{A_2} & \textcircled{A_3} & \\
S_3 & & & A_1 & A_2 & A_3 & \boxed{A_1} & \boxed{A_2} & \boxed{A_3}
\end{array}$$

The average latency $\mathcal{L}_{A^1}$ will be 2 because task A_4^1 can be initiated at $t = 7$. Clearly $\mathcal{L}_{A^1}$ is equal to the lower bound $\mathcal{L}_2$ of task A. Hence the throughput is maximum.

A value of the average latency of a modified task depends on a basic sequence that we use to modify a task. It can be seen, in general, that using smaller number of pseudo segments in a modified task will produce a smaller average latency. Hence it is important to find a basic sequence so that the number of pseudo segments in a modified task becomes minimum. To obtain such a basic sequence, we will employ a graph called a task graph which will be discussed next.

2.3. Task Graph

From a given task $\{S_1S_2\text{--}\}$, we will form an oriented graph [2] whose vertices are segments and edges indicate neighboring relationships of segments in the task. As an example, consider a task $A = \{S_1S_2S_3S_4S_5S_2S_4S_1S_5\}$. We will use a graph with five vertices S_1, S_2, S_3, S_4 and S_5. First we draw an edge from S_1 to S_2 because S_2 is next to S_1 in A as shown in Fig. 1a. Since after S_2 is S_3, we draw an edge from S_2 to S_3. Then we draw an edge from S_3 to S_4 and an edge from S_4 to S_5. Next we draw an edge from S_5 to S_2 and so on until we draw the last edge from S_1 to S_5. The resultant graph is shown in Fig. 1b which is a task graph corresponding to A.

We will search some properties of task graphs which lead to the development of a method of obtaining a basic sequence. Consider a task graph in Fig. 2a which corresponds to a task $A = \{S_1S_2S_3S_4S_5S_1S_3\}$. A basic sequence means a HD (Hamiltonian Directed) circuit C which passes each vertex in a task graph once. For convenience, we use a sequence of vertices to indicate a HD circuit. Let edge e be from S_i to S_j in a task graph G. Then we say that edge e is not in a HD circuit if S_j is not next to S_i in the HD circuit. For example, suppose we choose a HD circuit $C = \{S_1S_2S_3S_4S_5\}$ as shown in Fig. 2b. Then edge a is not in C.

Consider an edge as a rubber chord. Stretching an edge a and hooking it to vertex S_2, we will have a resultant graph as shown in Fig. 2c where every edge is in C. The corresponding task is $\{S_1S_2S_3S_4S_5S_1 \enclose{circle}{S_2} S_3\}$. We can see that hooking an edge to a vertex S_i corresponds to inserting a pseudo segment S_i into A.

Consider a task graph in Fig. 3a whose corresponding task $B = \{S_1S_2S_3S_4S_5S_1S_4\}$. Suppose we choose a HD circuit $C = \{S_1S_2S_3S_4S_5\}$ as shown in Fig. 3b. Then edge b is not in C. So we stretch edge b and hook it to vertices S_2 and S_3 as shown in Fig. 2e. The corresponding task B^1 is $\{S_1S_2S_3S_4S_5S_1 \enclose{circle}{S_2} \enclose{circle}{S_3} S_4\}$. Note that the number of pseudo segments in B^1 is the same as the number of vertices that we hook edge b to.

Consider a task graph in Fig. 4a where there is a directed circuit consisting of two edges. Such a directed circuit is called a "twin." Suppose we choose a HD circuit C as shown in Fig. 4a. In order to make the twin in C, we must hook edge a to S_{p+1}, --, and S_n and edge b to S_2, --- and S_{p-1}. The total number of vertices to which this twin is hooked is n-2 where n is the number of vertices in a task graph. Since a twin has two vertices, and C has n vertices, we can see that a twin must hook to n-2 vertices to make a twin in any HD circuit C.

Consider a task graph in 4b where there is a directed circuit consisting of one edge which is called a "self-loop." Let C in the figure be a chosen HD circuit. Then we must hook a self-loop to all vertices except one at which the self-loop is connected. Thus for any HD circuit, a self-loop must be hooked to n-1 vertices where n is the number of vertices in a task graph.

Thus we have the following lemma.

Lemma 1: Suppose there are k_t twins and k_s self-loops in a task graph G consisting of n vertices. Let G^1 be obtained from G by deleting all twins and all self-loops. Then the total number K of vertices which are hooked by edges so that all edges are in a HD circuit C is equal to

$$K = K_o + (n - 1)k_s + (n - 2)k_t$$

where K_o is the total number of vertices to which edges are hooked so that all edges in G^1 are in C.

Let $C = \{S_1S_2 \text{ --- } S_n\}$ be a HD circuit. Then $[S_p, S_q]$ of C is a part of sequence C starting with S_p and ending with S_q. For example, suppose $C = \{S_1S_2S_3S_5S_4\}$. Then $[S_3S_4]$ of C is $[S_3S_5S_4]$ and $[S_5S_2]$ of C is $[S_5S_4S_1S_2]$. We define $W_a(C)$ is as follows: suppose edge $a = \{S_rS_t\}$ which means that a is connected from S_r to S_t. Then

$$W_a(C) = [\text{number of vertices in } [S_r, S_t] \text{ of } C] - 2.$$

As an example, consider a task graph in Fig. 2a. Let a HD circuit C be $C = \{S_1S_2S_3S_4S_5\}$. Hence $W_a(C) = 1$. Note that $W_a(C)$ is the number of vertices to which edge a should be hooked in order that edge a will be in C.

By defining W(C) as

$$W(C) = \sum_{\substack{\text{all edges}\\ \text{not in } C}} W_p(C)$$

we have the following lemma.

Lemma 2: With a HD circuit C, the total number of pseudo segments necessary to modify a task by a basic sequence corresponding to C in W(C).

With this lemma, Lemma 1 can be restated as

Lemma 3: W(C) can be expressed as

$$W(C) = W(C)_{\text{ of } G^1} + (n - 2)k_t + (n - 1)k_s$$

With Lemma 3, we have an important theorem as

Theorem 1: Let G be a task graph of task A. The minimum number of pseudo segments necessary to modify task A to a repeating form is by the use of a basic sequence corresponding to a HD circuit C which gives the minimum W(C) of G^1 where G^1 is obtained from G by deleting all twins and all self-loops.

Example 1: Consider a task A which is

$$A = \{S_1S_2S_3S_1S_4S_1S_2S_2S_4S_3S_2\}$$

the corresponding task graph is shown in Fig. 5a. By deleting all twins and all self-loops, the resultant graph G^1 is shown in Fig. 5b. It is clear that a HD circuit $G = \{S_1S_2S_4S_3\}$ gives W(C) of G^1 minimum (W(C) of $G^1 = 0$). Hence by Theorem 1, task A should be modified by basic sequence $\{S_1S_2S_4S_3\}$ as

$$A^1 = \{S_1S_2\enclose{circle}{S_4}S_3S_1\enclose{circle}{S_2}S_4\enclose{circle}{S_3}S_1S_2\enclose{circle}{S_4}\enclose{circle}{S_3}\enclose{circle}{S_1}S_2S_4S_3\enclose{circle}{S_1}S_2\}$$

Note that there are two twins and one self-loop. Hence by Lemma 3, the number of pseudo segments in A^1 will be $2(n - 2) + (n - 1) = 7$.

2.4. Modified Tasks by One Basic Sequence

Suppose there are k different tasks. Then overlapping k task graphs of these tasks, we can obtain a graph called a "combined task graph." Let G_c be a combined task graph. Then by taking a HD circuit of G_c, we can modify these k tasks so that all of them are repeating form of one basic sequence.

Lemma 4: k different tasks modified by one basic sequence have the number of pseudo segments equal to

$$W(C) \text{ of } G_c^1 + (n - 2)N_t + (n - 1)N_s$$

where C is a HD circuit corresponding to the basic sequence, G_c^1 is obtained from a combined task graph by deleting all twins and all self-loops, N_t is the number of twins and N_s is the number of self-loops.

The proof of this lemma can be accomplished by the use of Lemma 3. This lemma leads to the following important theorem.

Theorem 2: Suppose a HD circuit C gives the minimum value of W(C) of G_c^1 among all HD circuits. Then the number of pseudo segments in tasks modified by the use of one basic sequence is the minimum when the basic sequence corresponds to C.

As an example, consider tasks $A = \{S_1S_2S_3S_1S_3\}$ and $B = \{S_1S_3S_1S_2S_3\}$. From a flow chart in Table 4 for the initiation of the sequence ABAB

Table 4. Initiation of Sequence ABAB

$$\begin{array}{ccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\end{array}$$
$$\begin{bmatrix}
A & & B & A & B & & \\
 & A & & & & B & \\
 & & A & B & A & & B
\end{bmatrix}$$

we can see that the average latency $\ell_{AB} = 2.5$. A combined task graph G_c is shown in Fig. 6a. By deleting all twins, we have G_c^1 shown in Fig. 6b. By choosing $C = \{S_1S_2S_3\}$, we can see that W(C) of G^1 is 0. Thus the number of pseudo segments in modified tasks becomes minimum by Theorem 2. The modified tasks are

$$A^1 = \{S_1S_2S_3S_1\enclose{circle}{S_2}S_3\}$$

$$B^1 = \{S_1\enclose{circle}{S_2}S_3S_1S_2S_3\}$$

Since A^1 and B^1 are the same as far as the reservation table is concerned, the average latency $\ell = 2$.

3. SEGMENTS WITH BUFFER REGISTERS

In this section, we will investigate a pipeline where each segment has an additional buffer register so that when a task T_i in a segment can not be transferred to a next segment because the segment S_i is needed by another task, the task T_i can be stored in the buffer register. Also we assume that a task in such a buffer register can be transferred to any segment in a pipeline. An important question which should be answered for such a pipeline is whether there exists a situation of collision of tasks in buffer registers or not. Surprisingly there will not be such a situation. However, transferring tasks in a pipeline must be chosen properly. The development of a suitable way of transferring tasks for preventing collision in a pipeline is the main objective of this section.

3.1. An Example for Segments with Additional Buffer Registers

Normally, there is one buffer register at the output of each segment called an "output buffer register" so that when the computation for a task is completed, the task will be at the output buffer register. It is no difficulty to make hardware so that a task can stay in the output buffer register if so desired until another task enters into the segment. Hence, we will assume that a segment can hold a task.

We assume that there is an additional buffer register called an "auxilliary buffer register" at each segment so that a task in the output buffer register can be transferred to the auxiliary buffer register for holding. Hence, if a segment is not used for computation at a time, it can hold two tasks one at its output buffer register and the other at its auxiliary buffer register. Even if the segment is used for computation, one task can be stored in its auxiliary buffer register. As an example, consider tasks $A = \{S_1S_2S_3S_1S_3\}$ and $B = \{S_1S_3S_3S_2S_1\}$. Suppose Task A_1 is initiated at $t = 1$, task B_1 is initiated at $t = 2$ and task A_2 is at $t = 3$. Also the next waiting task B_2 should be initiated at the earliest time possible. Then a flow chart should be as shown in Table 5,

Table 5. Initiation of Sequence ABAB

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
S_1	A_1	B_1	A_2	A_1	B_2	$\textcircled{B_2}$	A_2	$\boxed{B_1}$	A_3	B_3	A_4	A_3	$\boxed{B_2}$	B_4	A_5	A_4	$\boxed{B_3}$	B_5	$\boxed{B_4}$	A_5
			(B_1)		(A_1)	(A_1)	(B_2)	(A_2)			(B_3)	(A_4)			(B_4)	(A_5)	(A_4)		(B_5)	
S_2		A_1		A_2			B_1			A_3	B_2	$\textcircled{B_2}$	A_4	$\textcircled{A_4}$	B_3	$\textcircled{B_3}$	A_5	B_4		
																		(A_5)		
S_3			A_1	B_1	A_2	B_1	$\boxed{A_1}$	B_2	$\boxed{A_2}$	B_2	A_3	B_3	$\boxed{A_3}$	B_3	A_4	B_4	$\textcircled{B_4}$	$\boxed{A_4}$	A_5	B_5
					(B_1)	(A_2)			(B_2)				(B_3)							

where a task inside of a parenthesis means that the task is in the corresponding buffer register. Encircling a task indicates that the segment is employed as to store the task. Note that the efficiency index for this case is $\mathcal{J} = 2$. To initiate a new task, we use the following rule.

Rule for Initiation: Suppose the first segment of a waiting task is S_r. Then task T can be initiated whenever segment S_r can be made vacant.

As an example, consider Table 5. Initiation of task A_2 at $t = 3$ is valid because segment S_1 is neither necessary for computation of task B_1 nor necessary for storing task B_1 because the corresponding auxiliary buffer register U_1 can be used for storing task B_1. Similarly, at $t = 5$, we can initiate task B_2. However at $t = 6$, we can not initiate a new task because segment S_1 is used for storing B_2 and the corresponding auxiliary buffer register is used for storing task A_1. At $t = 7$, a new task can not be initiated because task A_2 requires segment S_1 for computation and so on.

A question to be answered is whether the rule of initiation given previously is sufficient to prevent any collision in a pipeline. To show there is a way of transferring tasks without developing collision as long as the rule of initiation is obeyed, we will employ a graph called a "task flow graph."

3.2. Task Flow Graph

The locations of tasks in a pipeline at time t and the next segment to which each task hopes to transfer at time t + 1 is indicated by an oriented graph F called a task flow graph for time t as follows [2]:

(1) Each vertex of a task flow graph F represents either a segment or an auxiliary buffer register in a pipeline.

(2) If there is a task T_i in an auxiliary buffer register U_i at time t and the next segment that T_i should go to is S_j, then we place an oriented edge from vertex U_i to vertex S_j.

(3) If there is a task T_p in a segment S_p and the next segment that T_p should go to is S_q, we place an oriented edge from S_p to S_q. Also an oriented edge from S_p to U_p if there is a task in U_p.

As an example, a task flow graph F at time t = 7 in Table 5 is shown in Fig. 7. Note that task A_1 will not be shown in the task flow graph because A_1 will leave from a pipeline at t = 8.

Let edge e, be from a vertex V_1 to a vertex V_2 in a task flow graph. A "forward trace starting with edge e," is a sequence of alternation of a vertex and an edge starting from V_1, then e_1 and V_2 and so on such that 1 the orientation of every edge in the sequence is from the preceding vertex to the succeeding vertex, 2 no vertices in the sequence appear more than once except the last vertex which can be the same as one already in the sequence and 3 the sequence will terminate only if either there is no edge connected from* the last vertex or the last vertex has already been in the sequence. As an example, consider the task flow graph in Fig. 8. One forward trace starting with edge e_1 is $\{S_1e_1S_2e_2S_3e_3U_3e_4S_2\}$ which is terminated at vertex S_2 because S_2 appeared already in the sequence. Another forward trace starting with e_1 with e_1 is $\{S_1e_1S_2e_6U_2e_7S_1\}$. The other is $\{S_1e_1S_2e_2S_3e_5S_5\}$ which is terminated at S_5 because there is no edge from S_5.

A "backward trace starting with an edge" is the same as a forward trace except we will form a sequence of alternation of a vertex and edge backward. As an example, a backward trace starting with edge e_3 in Fig. 8 can be $\{S_2e_6U_2e_7S_1e_1S_2e_2S_3e_3U_3\}$.

We say that a forward trace is a "forward task trace" whenever auxiliary buffer register U_p of segment S_p in the trace is occupied by a task, then U_p is also in the trace. Similarly we say that a backward trace is a "backward task trace" whenever auxiliary buffer register U_p of a segment S_p in the trace is occupied by a task, then U_p is also in the trace.

The forward trace $\{S_1e_1S_2e_2S_3e_3U_3e_4S_2\}$ in Fig. 8 is not a forward task trace. The forward trace $\{S_1e_1S_2e_6U_2e_7S_1\}$ is a forward task trace. The backward trace $\{S_2e_6U_2e_7S_1e_1S_2e_2S_3e_3U_3\}$ is a backward task trace.

A "task Circuit C" is a directed circuit which satisfies that auxiliary buffer register U_p of a segment S_p in C is occupied by a task, then U_p must be in C. As an example, the forward task trace $\{S_1e_1S_2e_6U_2e_7S_1\}$ in Fig. 8 is a task circuit. Now we are ready to show that the rule of initiation is enough to avoid collision by a proper transfer of tasks in a pipeline.

Theorem 3: There exists a way of transferring tasks in a pipeline such that the rule of initiation is sufficient to avoid any collision in a pipeline.

---

*An edge is from a vertex V if the edge is connected at V and its orientation is away from V.

Proof: We will prove the theorem by showing that there is a way of transferring tasks in a pipeline without collision. Suppose there exists no collision at time t. In a task flow graph F for time t, suppose there exists a vertex S_p whose incoming degree $d^-(S_p) > 1$*.

Case I: Suppose there is a task in U_p. Let e_p be an edge leaving at U_p. Note that there is one edge leaving from every U_r containing a task. Also there is one edge from S_r to U_r if a task is in U_r. Furthermore there is exactly one edge leaving from S_r if a task is in S_r but no task is in U_r. Thus there is exactly one forward task trace starting with e_p.

If this forward task trace is a task circuit, then we transfer tasks in the vertices of the circuit for t + 1. All other tasks stay in the same place. This transfer clearly will not produce collision. Also if the first segment S_w of the waiting task is either vacant or corresponding buffer register U_w is vacant, the waiting task can be initiated at t + 1.

Suppose a forward task trace is terminated because the last vertex has already appeared previously as shown in Fig. 9. Suppose the vertex v following S_r is corresponding auxiliary buffer register U_r. Then it is clear that S_r has $d^-(S\ ) > 1$, and trace circuit C^1 in Fig. 9 is a desired one. That is we transfer tasks in vertices in C^1 for t + 1.

Suppose vertex V is not U_r. Hence U_r is vacant. Let P be a portion of the forward task trace from U_p to S_r. We transfer tasks in vertices in P and transfer a task in S_r to U_r for t + 1. Let S_w be the first segment of a waiting task. Then if either segment S_w is vacant or U_w is vacant, the waiting task can be initiated at t + 1.

Suppose the forward task trace is terminated because there is no edge leaving from the last vertex S_2. Then the last vertex S_2 is either vacant or will be vacant at t + 1. Thus we will transfer tasks in the vertices of the forward task trace for t + 1 which will not produce collision. Again a waiting task can be initiated if the rule of initiation is satisfied.

Case II: Suppose U_p is vacant. Then we will use a backward task trace starting with an edge entering at S_p and the results will be the same as those of Case I.

When every vertex has incoming degree of one or zero in F, we can transfer all tasks without collision. This concludes the proof.

Q. E. D.

4. COMPARISON OF THREE METHODS

We have developed two methods, one of which is to modify tasks into repeating form of one basic sequence and the other is to employ auxiliary buffer registers. These two and a method of scheduling initiation will be compared here.

Let tasks A and B be

$$A = \{S_1S_2S_3S_4S_1S_3S_2S_4\}$$

---

*Incoming degree $d^-(S_p)$ is the number of edges entering at S_p.

and

$$B = \{S_3S_1S_3S_4S_3S_2S_4S_1\}.$$

the initiating sequence ABAB --- by scheduling is shown in Table 6 which shows that the average latency $\mathcal{L}_{AB} = 11/2$.

$$\begin{array}{c|ccccccccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & \\
S_1 & A & & & & A & & & B & & & & A & & \boxed{B} & \\
S_2 & & A & & & & & A & & & & & B & A & & \\
S_3 & & & A & & & A & B & & B & & B & & & A & \\
S_4 & & & & A & & & & \boxed{A} & & B & & & B & & A
\end{array}$$

In order to find an optimum basic sequence, we form a combined task graph of A and B as shown in Fig. 10a. Then delete all twins, the resultant graph G^1 is shown in Fig. 10b. A HD circuit which minimizes W(C) is shown in the figure. Hence an optimum basic sequence will be $\{S_1S_3S_2S_4\}$. With this basic sequence, tasks will be modified as

$$A' = \{S_1 \textcircled{S_3} S_2 \textcircled{S_4}\textcircled{S_1} S_3 \textcircled{S_2} S_4S_1S_3S_2S_4\}$$

and

$$B' = \{S_3 \textcircled{S_2}\textcircled{S_4} S_1S_3 \textcircled{S_2} S_4 \textcircled{S_1} S_3S_2S_4S_1\}$$

The flow chart becomes as shown in Table 7.

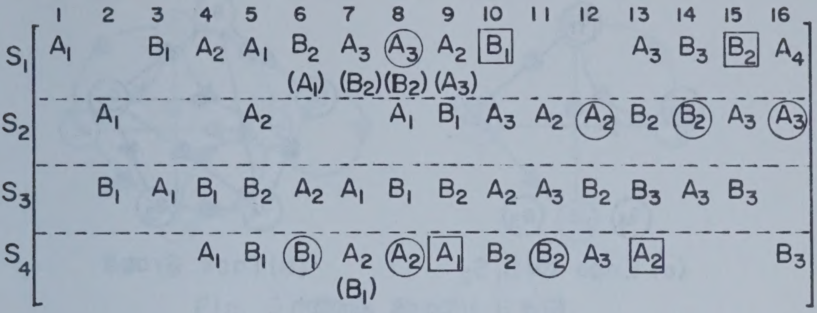
Table 7. Initiation of $\{A'B'A'B'\}$

$$\begin{array}{c|ccccccccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\
S_1 & A'_1 & & & A'_2 & \textcircled{A'_1} & B'_1 & B'_2 & \textcircled{A'_2} & A'_1 & \textcircled{B'_1} & \textcircled{B'_2} & A'_2 & A'_3 & \boxed{B'_1} & \boxed{B'_2} \\
S_2 & & & A'_1 & \textcircled{B'_1} & \textcircled{B'_2} & A'_2 & \textcircled{A'_1} & \textcircled{B'_1} & \textcircled{B'_2} & \textcircled{A'_2} & A'_1 & B'_1 & B'_2 & A'_2 & \\
S_3 & & \textcircled{A'_1} & B'_1 & B'_2 & \textcircled{A'_2} & A'_1 & B'_1 & B'_2 & A'_2 & A'_1 & B'_1 & B'_2 & A'_2 & & \\
S_4 & & & & \textcircled{A'_1} & \textcircled{B'_1} & \textcircled{B'_2} & \textcircled{A'_2} & A'_1 & B'_1 & B'_2 & A'_2 & \boxed{A'_1} & B'_1 & B'_2 & \boxed{A'_2}
\end{array}$$

Note that A'_3 can enter at t = 13. Hence the average latency $\mathcal{L}_{A'B'A'B'} = 3$ which is a great improvement over that with unmodified tasks.

With auxiliary buffer registers, a flow chart for initiation of ABAB --- is shown in Table 8.

Table 8. Initiation of ABAB -- With Buffer Registers



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S_1	A_1		B_1	A_2	A_1	B_2 (A_1)	A_3 (B_2)	$\enclose{circle}{A_3}$ (B_2)	A_2 (A_3)	$\boxed{B_1}$			A_3	B_3	$\boxed{B_2}$	A_4
S_2		A_1			A_2			A_1	B_1	A_3	A_2	$\enclose{circle}{A_2}$	B_2	$\enclose{circle}{B_2}$	A_3	$\enclose{circle}{A_3}$
S_3		B_1	A_1	B_1	B_2	A_2	A_1	B_1	B_2	A_2	A_3	B_2	B_3	A_3	B_3	
S_4				A_1	B_1	$\enclose{circle}{B_1}$	A_2 (B_1)	$\enclose{circle}{A_2}$	$\boxed{A_1}$	B_2	$\enclose{circle}{B_2}$	A_3	$\boxed{A_2}$			B_3

We can see that the average latency $\mathcal{l}_A \cong 2.5^+$.

5. CONCLUSION

It is easily seen that tasks modified into a repeating form of one basic sequence are easiest to initiate without any collision. Also if we know a probability distribution of tasks, we can modify a combined task graph by using the probability distribution of each task as a weight of edges of a task graph for the task so that a basic sequence obtained from a combined task graph will modify tasks to have the lowest average latency under the given probability distribution.

With auxiliary buffer registers the rule for initiation is simple. However, proper transfer of tasks in a pipeline is essential in order to avoid collision. However, the control for a proper transfer of tasks in a pipeline may not be as complicated as that used for a method of optimum scheduling. If one sacrifices a few percent of the average latency, to make a rule of initiation more restricted, one may have a simpler transfer of tasks in a pipeline.

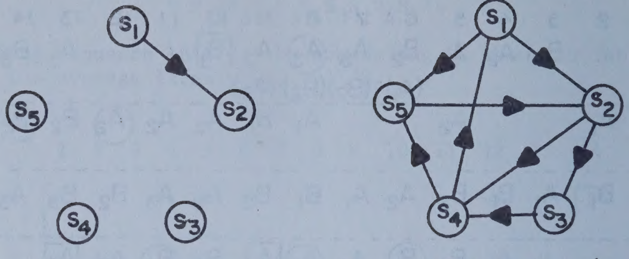
I would like to acknowledge the valuable help which I received from discussions with Professor Edward S. Davidson and Dr. Janak Patel on this subject.

REFERENCES

[1] A. T. Thomas and E. S. Davidson, "Scheduling of Multiconfigurable Pipeline," Proc. 12th Allerton Conf. on Circuit and System Theory, pp. 658-669, Oct. 1974.

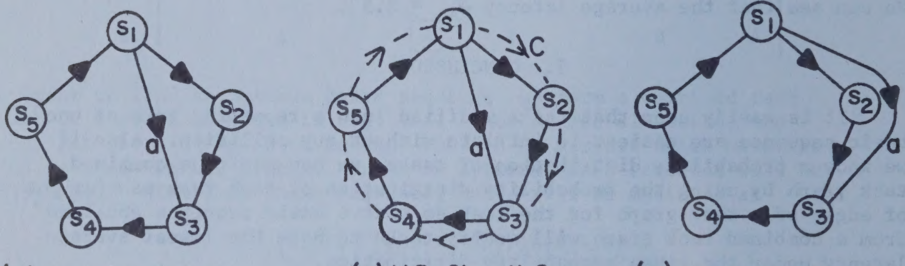
[2] W. Mayeda, Graph Theory, John Wiley and Sons, New York, 1972.

[3] E. S. Davidson, "Scheduling for Pipeline Processors," Proc. 7th Hawaii Int. Conf. on System Science, Jan. 1974.


S1
S2
S3
S4
S5

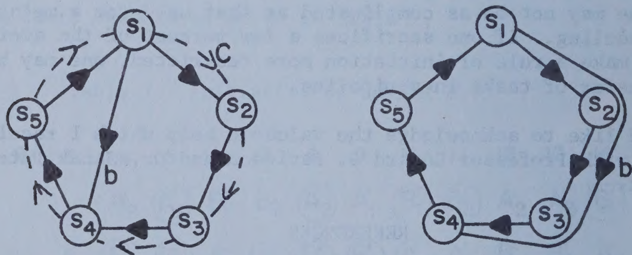
(a) Edge for $S_1 S_2$ (b) Task Graph

Fig. 1 Task Graph


S1
S2
S3
S4
S5
a
C

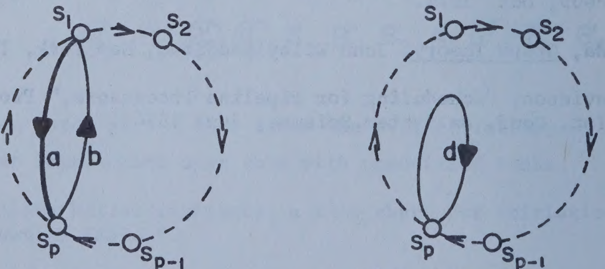
(a) Task Graph for A (b) HD Circuit C (c) Hooking Edge a

Fig. 2 Modifying Task Graph


S1
S2
S3
S4
S5
b
C

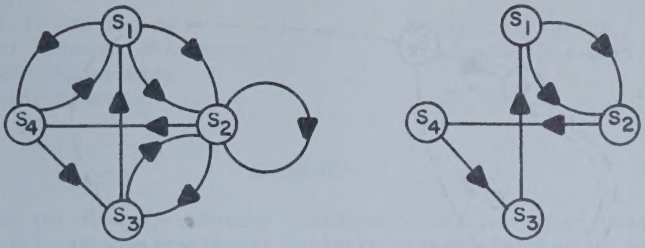
(a) Task Graph for B (b) Hooking Edge b to S_2 & S_3

Fig. 3 Modifying Task Graph


S1
S2
Sp
Sp-1
a
b
d

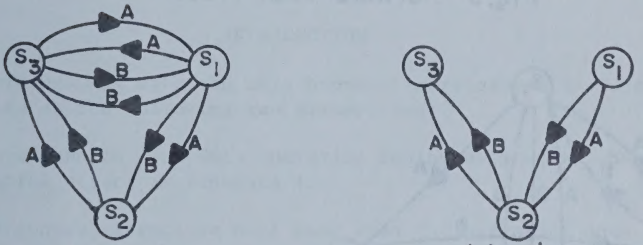
(a) Twin (b) Self-loop

Fig. 4 Twin & Self-loop


S1
S2
S3
S4
S1
S2
S3
S4

(a) G (b) G'

Fig. 5. Task Graphs G & G'


S3
S1
S2
A
A
B
B
A
B
B
A
S3
S1
S2
A
B
B
A

(a) G_c (b) G'_c

Fig. 6. Combined Task Graphs

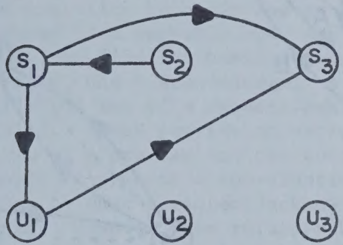

S1
S2
S3
U1
U2
U3

Fig.7 Task Flow Graph

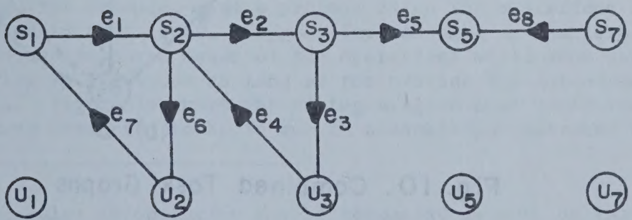

S1
e1
S2
e2
S3
e5
S5
e8
S7
e7
e6
e4
e3
U1
U2
U3
U5
U7

Fig. 8. Task Flow Graph

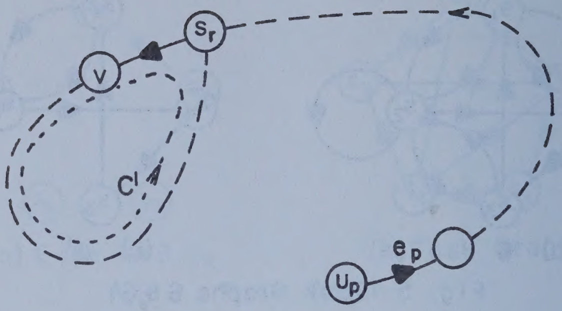
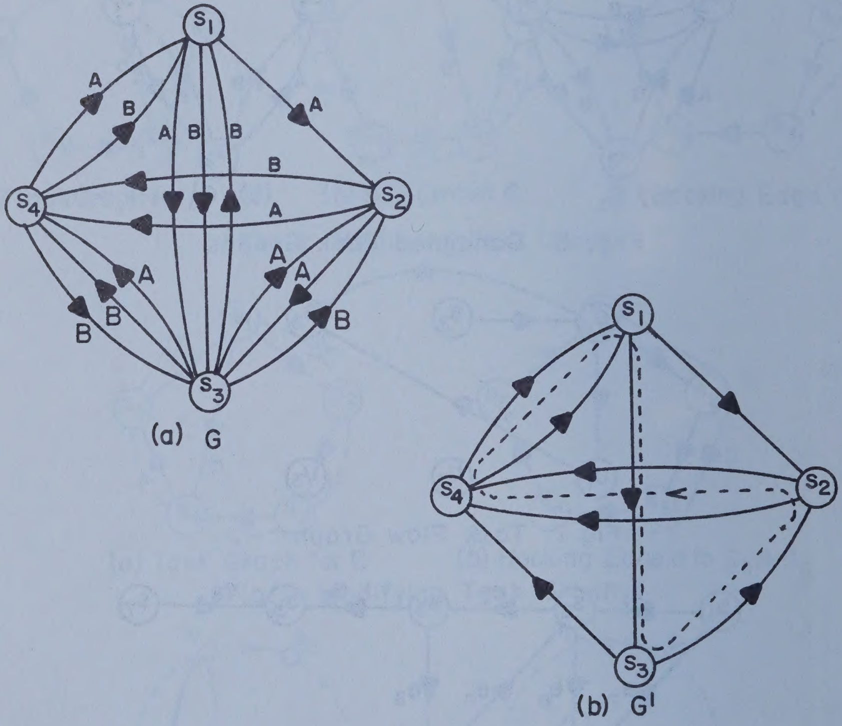


Fig.9 Forward Task Trace


S1
A
B
A
B
B
A
B
S4
S2
A
A
A
A
B
B
B
S3
S1
S4
S2
S3

(a) G

(b) G'

Fig. 10. Combined Task Graphs

ASYNCHRONOUS COORDINATION OF DATA OPERATIONS: A SCHEME EXPLOITING OVERLAP POTENTIAL*

MISHELL J. STUCKI
Computer Systems Laboratory
Washington University
St. Louis, Missouri

MAURICE L. PEPPER
St. Louis, Missouri

ABSTRACT

A scheme for the asynchronous coordination of data operations is presented. This scheme is similar to other schemes in that it uses a request/acknowledge protocol and uses a common sequencing rule. However, the relationship of the request and acknowledge to the operation execution is substantially different, and this makes it possible to exploit the overlap potential of a process.

INTRODUCTION

Most work dealing with the asynchronous coordination of data operations has been based on the following two assumptions:

1. The execution of a data operation begins as soon as the control structure requests it.
2. The control structure must know when the operation ends in order to be able to sequence subsequent operations.

These assumptions have naturally led to a request/acknowledge protocol in which a request causes execution to begin and an acknowledge indicates its end. Although this scheme is a workable one, and systems have been successfully designed around it, we find the basic assumptions and the consequent relations (request = begin, end = acknowledge) to be unnecessary. Our quarrel here is not with the use of a request/acknowledge protocol nor with the sequencing rule[1] used. These provide an extremely convenient mapping between the flow diagram of a process and the control structure that coordinates it. In this paper we present a coordination scheme that is similar to other schemes in that it uses a request/acknowledge protocol and uses the same sequencing rule. However, the relationship of the request and acknowledge to the operation execution is substantially different, and this makes it possible to exploit the overlap potential of a process.

Suppose, for example, that a process calls for operations X, Y, and Z to be executed in that order. Suppose further that an analysis of the data dependencies and resource needs of the operations would show that they could safely overlap in execution so long as the overlap did not exceed that shown in Figure 1a. Figure 1b shows the timing diagram that would result if a request caused execution to begin and an acknowledge indicated the end of

---

[1] Sequencing Rule: An operation may be requested as soon as the previous operation in the sequence has been acknowledged.

* This work has been supported by the Division of Research Resources of the National Institutes of Health under Grant No. RR-00396.

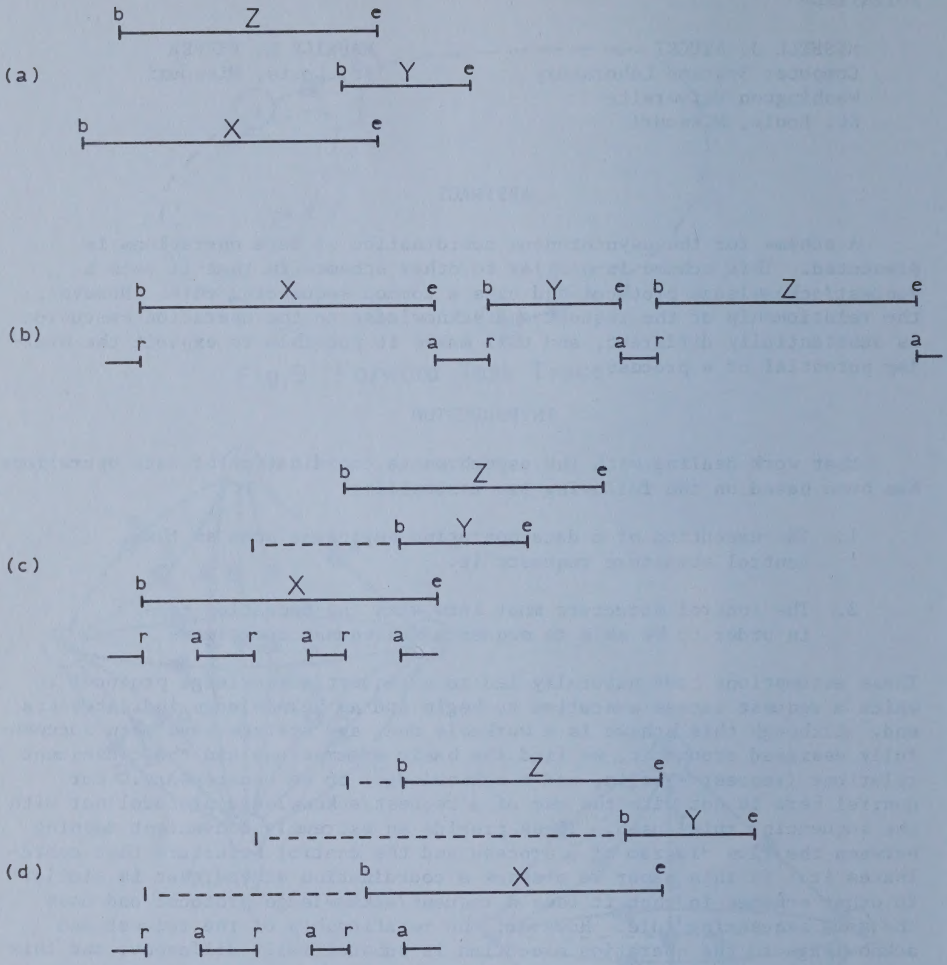

(a)
b Z e
b Y e
b X e
(b)
b X e b Y e b Z e
r a r a r a
(c)
b Z e
b Y e
b X e
r a r a r a
(d)
b Z e
b Y e
b X e
r a r a r a

Figure 1. The sequence X, Y, Z. (a) Maximum safe operation overlap. (b) Timing diagram if request = begin and end = acknowledge. (c and d) Timing diagrams possible with the new scheme.

execution. Interpretation of the diagram is straightforward except possibly for the a-r intervals. These intervals account for the propagation times of the acknowledge and request signals and also account for time spent in intervening control steps. Figure 1c shows a timing diagram that could result with our scheme. Here, an acknowledge occurs a short time after the request, and its occurrence is totally unrelated to operation execution state. The marked intervals (⊢——⊣) in this diagram are the same as in Figure 1b but are differently arranged. Note that operation Y is requested before it can safely begin. This situation is automatically detected by logic in the Y operation unit and execution of Y is postponed until the appropriate time. Note also that operation Z begins before Y even though Y was requested first. Execution order is being dictated more by the constraints reflected in Figure 1a than by the order of request. Actually, with this scheme, it is even possible to achieve maximum overlap, as illustrated in Figure 1d. That figure assumes that all three operations are requested before it is safe for them to begin, and as a result, each begins execution at the earliest possible moment. Of course, in practice, either of the situations diagrammed in Figures 1c and 1d could occur, or something intermediate. It all depends on the relationship of operation X to the operations that precede it. In every case, though, the performance is much better than that shown in figure 1b, and this results from the fact that the data operations overlap with each other and with the a-r intervals.

This scheme depends on the inclusion of certain logic in each data operation unit. This logic automatically detects unmet data dependencies and unavailable resources and keeps a requested operation from beginning when such conditions are present.

THE SCHEME

Data variables in our system are represented by a four-tuple called a data cell. The elements of a data cell are:

[variable value, sequence no., program no., status bit].

The sequence number is zero when the system starts and is automatically indexed each time the variable value is replaced. The sequence number thus indicates whether the value held by the variable is the zero-th, tenth, sixteenth, etc.. The function of the program number and status bit will be discussed later.

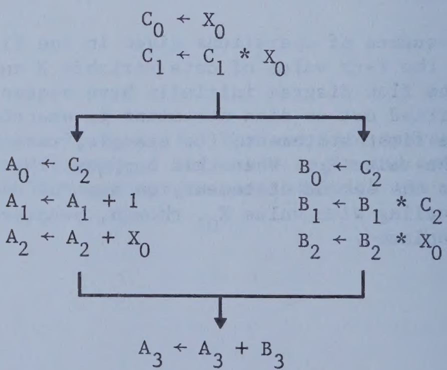
Consider the sequence of operations given in the flow diagram below. If we let X_i denote the i-th value of data variable X and assume that all data variables in the flow diagram initially have sequence number zero, then the process carried out as each statement is executed is that shown in parentheses. The first statement, for example, results in the value C_0 being replaced by the value X_0. When this happens, the value C_1 is created. Thus, when we get to the second statement, we are now dealing with value C_1. We are still dealing with value X_0, though, because the original value of X has not been replaced.

$C \leftarrow X$	$(C_0 \leftarrow X_0)$
$C \leftarrow C * X$	$(C_1 \leftarrow C_1 * X_0)$
$A \leftarrow C$	$(A_0 \leftarrow C_2)$
$B \leftarrow C$	$(B_0 \leftarrow C_2)$
$B \leftarrow B * C$	$(B_1 \leftarrow B_1 * C_2)$
$B \leftarrow B * X$	$(B_2 \leftarrow B_2 * X_0)$
$A \leftarrow A + 1$	$(A_1 \leftarrow A_1 + 1)$
$A \leftarrow A * X$	$(A_2 \leftarrow A_2 * X_0)$
$A \leftarrow A + B$	$(A_3 \leftarrow A_3 + B_3)$

We shall refer to statements such as $C \leftarrow C * X$ as operation names, while we will call statements such as $C_1 \leftarrow C_1 * X_0$ instructions. An instruction contains an operation name plus the sequence numbers of the data values being used and replaced.

Suppose that we had enough processors (data operation units) so that each statement in the flow diagram could be executed by a different processor. Suppose, further, that as each statement is about to be executed, the processor creates and saves within itself an instruction describing the action about to be performed. This is easily done because the processor certainly knows the operation name, and the sequence numbers for the data values can be copied from the data cells. For example, when it is time to execute the fifth statement in the sequence, the sequence number in data cell B is 1 and the sequence number in data cell C is 2. Thus, before beginning execution of the operation $B \leftarrow B * C$, the processor stores away the instruction $B_1 \leftarrow B_1 * C_2$.

Now after having completed execution of the sequence, suppose that we were to have the processors execute the operations again. This time, however, we shall not cause the processors to execute the operations in the order of the flow diagram. Instead, we shall let each processor execute the instruction stored within it, the basic rule being that the instruction is to be executed as soon as the appropriate data values are available. The execution ordering that would occur is shown in the diagram below.


$C_0 \leftarrow X_0$
$C_1 \leftarrow C_1 * X_0$
$A_0 \leftarrow C_2$
$A_1 \leftarrow A_1 + 1$
$A_2 \leftarrow A_2 + X_0$
$B_0 \leftarrow C_2$
$B_1 \leftarrow B_1 * C_2$
$B_2 \leftarrow B_2 * X_0$
$A_3 \leftarrow A_3 + B_3$

This sequence gives the same final values for variables A, B, and C, but performs the actions of the original sequence concurrently wherever possible. This execution ordering is the essential ordering imposed by the data dependencies, and a higher degree of concurrency cannot be obtained without altering the original set of statements.

The first time that we executed the sequence described by the original flow diagram, we carried out two processes: the execution of the data operations and the programming of the processors. We now show that the programming can be done without executing the data operations. Let us assume for the moment that a processor has a programming mode such that, if we set the processors in the system into this mode, execution requests will be interpreted as programming requests. Now, we know that in order to create an instruction, we need the operation name and the sequence numbers of the data values. As before, the processor knows the operation name, but now it cannot get the sequence number information that it needs from the sequence numbers in the cells. This results from the fact that the sequence numbers are permanently tied to the data values, and since new data values are not being created, the sequence numbers in the cells remain 0. We solve this problem by making the cell entry called program number supply the needed information. This is done by making all program numbers 0 when the system starts, and by indexing the program number in a given cell each time an instruction is created which, if executed, would cause the sequence number in that cell to be indexed. Thus, as we program successive statements in the flow diagram, the program numbers in the cells have the sequence number information needed for each instruction. As a result, the creation of an instruction now consists of the following two steps.

1. Save the instruction name and the program numbers of the data variables.

2. If the instruction specifies the replacement of a data value, cause the program number of that data variable to be indexed.

The programming process just described uses only the program number of a cell, and the instruction execution process described before that uses only the data value and sequence number of a cell. It is thus possible for both processes -- instruction creation and instruction execution -- to operate concurrently. By this we mean that instruction execution will automatically take place as soon as the appropriate data values are available. An instruction may thus be executed as soon as it is created or much later. It all depends on the data dependencies of the instruction.

In our system, programming is controlled by the system control structure. A request signal causes a processor to create and store within itself an instruction, and the processor issues an acknowledge signal as soon as this has been accomplished. Thus, referring back to Figures 1c and 1d, the short interval shown between the arrival of a request and the subsequent issuance of an acknowledge represents the time required to program the operation. Operation execution automatically begins as soon as the needed data values are available.

FACTORS AFFECTING PERFORMANCE

There are several factors that affect the performance of the basic scheme, and most of them relate to the amount of logic one is willing to include in a processor or to the number of processors in the system. The effect of some of these is much greater than one might expect, as we shall see shortly. Perhaps the most interesting consideration is the effect of having or not having data buffers at the processor inputs. Let us assume that a processor has an input buffer for each input variable and that, if an instruction requires a specific data value, the value will automatically be copied into the input buffer as soon as it is present in the cell. To see the type of performance improvement that this buffering scheme offers, suppose that the three instructions ($A_7 \leftarrow B_2 * C_3$), ($C_3 \leftarrow B_2 + 1$), and ($B_2 \leftarrow 0$) have been programmed into separate processors. Figure 2 shows the execution timing that could occur. In this Figure, time increases from left to right and the vertical lines indicate the time of occurence of the values B_2, C_3, and A_7. A horizontal line shows the manner in which an instruction execution takes place. The fact that the lines are initially broken indicates that the processors involved already hold the instructions shown and are waiting for the data values to arrive. Figure 2a shows the execution timing given the input buffering scheme, and Figure 2b shows the

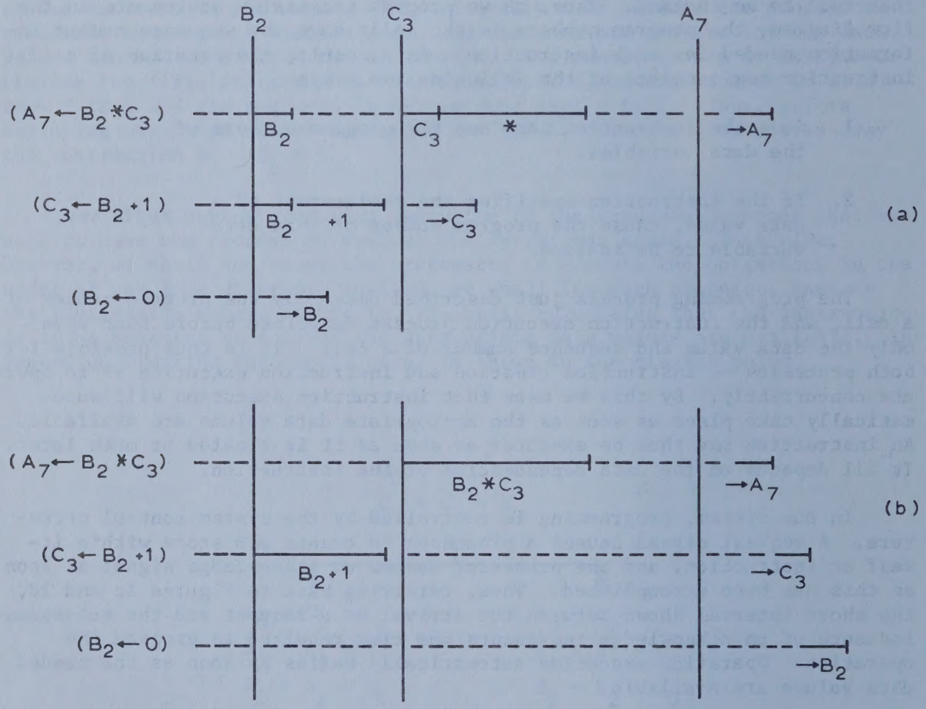


Figure 2. Execution timing for three instructions. (a) With input buffers. (b) Without buffers.

timing given no buffers. The top line of Figure 2a is interpreted as follows. Nothing happens until the value B_2 occurs, at which time a small amount of time is spent copying B_2 into the processor buffer. After that, nothing more happens until the value C_3 occurs. C_3 is then copied into the processor buffer and the multiplication operation takes place. After the product is formed, nothing more happens until the value A_7 occurs and has been copied into the buffers of all processors that need it. Then the value calculated by this instruction replaces A_7. The execution timing of the other two instructions shown in Figure 2a is interpreted in a similar fashion. In Figure 2b, the top line shows that nothing happens until both B_2 and C_3 are present. Then the multiplication operation takes place. After the product is formed, nothing more happens until A_7 occurs and has been used by all operations that need it. Then the value calculated by this instruction replaces A_7. The other two lines are interpreted in a similar fashion. Note that the absence of buffers means that the value C_3 cannot be replaced until the instruction that uses C_3 is completed. Similarly, the value B_2 cannot be replaced until the other two instructions that use B_2 are completed. The buffered scheme has no such problem because a processor is dependent on the value in a cell for only a small amount of time. Consequently, the last two operations finish earlier and subsequent operations that depend on B and C can begin sooner. It is clear from this example that the inclusion of input buffers can improve system performance. The degree of improvement depends on many factors and is a topic beyond the scope of this paper.

A second factor affecting performance is programming speed. This puts an absolute limit on system performance since an operation cannot occur unless it has been programmed. While it is true that the steps involved in creating an instruction are few and simple, they do take some time, and system design should focus on keeping this time as small as possible. A third factor is the number of processors in the system. It is highly unlikely that there are so many processors that a processor will only execute one operation during the running of the system. A given processor will probably be called upon many times, and this can result in the following situations:

1. A processor is asked to program an instruction before it has executed a previously programmed instruction. If a processor can only store one instruction, then the processor must postpone honoring the request until it has executed the stored instruction. This slows down the programming phase of the system and may, as a result, affect system performance. The probability of such a situation occurring can be substantially reduced by giving processors the ability to store several instructions. If one were to take into account all other system factors, one would probably find that a very small amount of storage, say three or four instructions worth, would essentially eliminate the problem.

2. A processor contains two instructions that could be executed concurrently. When this occurs, the instructions will be executed one at a time, and thus the potential concurrency of the instructions will not be realized. Such situations can be avoided in system sequences that are process invarient by careful analysis at design time. When sequences are process dependent, avoiding such situations is difficult and is a topic beyond the scope of this paper.

A fourth consideration relates to conditional branch operations. Since these determine the subsequent operations that are to be programmed, the programming process cannot proceed past such an operation until the outcome of the operation is known. As a result, these operations limit the extent to which the programming process can get ahead of the execution process. This in turn can limit the amount of potential concurrency that is realized. We cannot, at this time, suggest a way of avoiding this.

ADDITIONAL SCHEME DETAILS

The following details are necessary to the proper functioning of the system. They were omitted in previous discussions because they would have complicated the discussion at the time.

1. The status bit. The purpose of the status bit is to keep a data value from being replaced prematurely. For example, consider the following sequence and the associated instructions given in parentheses.

$$A \leftarrow B * C \qquad (A_0 \leftarrow B_0 * C_0)$$
$$C \leftarrow B + 1 \qquad (C_0 \leftarrow B_0 + 1)$$
$$B \leftarrow 0 \qquad (B_0 \leftarrow 0)$$

In this example, we see that the replacement value for B_0 requires no calculation and the replacement could thus be made as soon as the system starts. However, if this actually happened, it would interfere with the first two operations in our sequence and wrong values would result. We use the status bit to prohibit such premature replacement. Specifically, there is a system rule that prescribes that a data value may only be replaced if the status bit is 1. When the system starts, the status bit has value 0 and thus no replacements can be made. This situation persists until all operations that need the initial data value have been programmed, a fact indicated by the program number being larger than the sequence number. When this condition occurs, each processor connected with the cell sends a cell a signal as soon as it is through with the present data value. When a signal has been received from each processor, it is then permissable for the data value to be replaced, and the status bit becomes 1. Each time replacement occurs, the status bit becomes 0, and the process just described repeats.

2. Sequence number and program number considerations. These numbers have the same range of values and they automatically wrap around when the range is exceeded. Although we have previously specified that both numbers have initial value 0, they may have any initial value so long as they have the same value. The fact that an instruction must be created before it can be executed means that, conceptually, the program number is always larger than or equal to the sequence number. As a result, the status bit logic can assume that the program number is larger if the two numbers differ in value. The finite range of the numbers and the wrap-around feature make it possible for the program number to move so far ahead of the sequence number as to become equal to it. Should this occur, new instructions would execute prematurely and the status logic would become confused. If it is felt that this is a real possibility in a system, then the programming logic within each processor must be designed so that it will not index a program number until it can do so without making that number equal to the sequence number.

CONCLUSION

We have presented a scheme for the asynchronous coordination of data operations which exploits the overlap potential of a process. The basic idea has been to separate sequence specification from operation execution so that an operation does not have to execute in the order specified. Our description has been very general and not meant to suggest a specific application or method of realization. This is intentional because we feel that the scheme can be applied at many levels of system design and in many types of systems. In each case, implementation details would depend on the specific requirements of the application. In the section on performance, our discussion of the merits of buffering was not carried as far as it might have been. There are also advantages to be gained by having output buffers, and, if other system factors are ideal, in having input queues and output queues. We felt that these extensions would be obvious enough given the initial idea.

SYNTHESIS OF LEVEL SEQUENTIAL CIRCUITS: FURTHER DEVELOPMENT OF A PROCEDURE BASED ON A PETRI-NET TYPE OF BEHAVIORAL DESCRIPTION

MISHELL J. STUCKI
Computer Systems Laboratory
Washington University
St. Louis, Missouri

ABSTRACT

This paper presents further developments in a synthesis procedure for level sequential circuits which derives circuit equations directly from a Petri-net type of behavioral description. First, a method of net simplification is given which is useful in reducing the number of secondary variables in the solution. Second, a method is given for handling the non-trivial arbitration situation modeled by simultaneously enabled transitions with a common input place.

INTRODUCTION

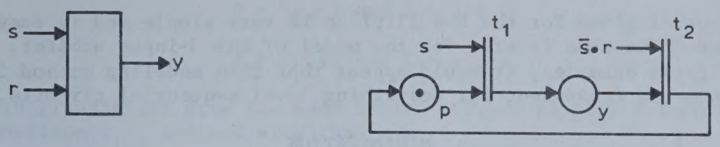
The synthesis approach presented in [1] gave a procedure for deriving circuit equations directly from a Petri-net type of behavioral description. Although the approach was developed for synthesizing transition-logic circuits, it works for level sequential circuits in general. In this paper, we demonstrate this fact and introduce two further developments, a method of net simplification useful in reducing the number of secondary variables in the solution, and a method of handling the non-trivial arbitration situation that is modeled by simultaneously enabled transitions with a common input place.

MODELING

Our first task is to show that the Petri-net type of behavioral description presented in [1] can be used to describe a level sequential circuit without requiring that the circuit behavior be expressed in terms of transition signals. We demonstrate this fact by example, beginning with the simplest of level sequential devices, the R-S flipflop, and then moving on to a more complex device, a 3-input arbiter.

Figure 1a shows the black box representation of an R-S flipflop with set input s, reset input r, and output y. The output value is set to 1 if s=1 and is reset to 0 if r=1. If both r and s are 1, the set input wins. Figure 1b shows our model for this behavior. Place y represents output y and a token in place y represents output value 1. Since place y is not shown holding a token, the initial output value is 0. Place y receives a token when transition t_1 fires, and, since place p has an initial token, this firing occurs as soon as s=1. The input condition s=1 thus sets y to 1. Once place y gets a token, the token remains until transition t_2 fires, and that event occurs as soon as we have the input conditions s=0 and r=1. The condition r=1 thus resets y to 0, but only if s≠1.

This work has been supported by the Division of Research Resources of the National Institutes of Health under Grant No. RR-00396.

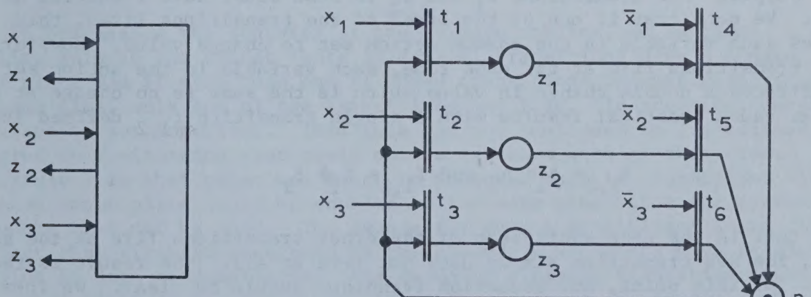


(a) (b)

Figure 1. The R-S flipflop. (a) Representation. (b) Model.

Figure 2a shows the black box representation of a 3-input arbiter. This device is used in systems to control access to a resource which, though shared by three concurrent processes, must only be used by one process at a time. In order to gain access to this resource, a process must request and obtain permission from the arbiter. Process i communicates with the arbiter via input x_i and output z_i. Permission is requested by making $x_i=1$ and the arbiter grants permission by making $z_i=1$. When the process finishes with the resource, it makes $x_i=0$ and the arbiter then terminates permission by making $z_i=0$. The arbiter grants permission to each process that requests it, but does so one at a time. Once permission is granted, other requests are ignored until that permission terminates. In the case of simultaneous or overlapping requests, the order in which permission is granted is arbitrary.

Figure 2b shows our model for this behavior. The places labeled z_i represent the device outputs and a token in these places indicates the output value 1. A token in place p asserts the condition "access available." In the initial state (that shown), all input and output lines are 0 and access is available. When an x_i input becomes 1, transition t_i fires and moves the token from place p to place z_i. The token in place z_i gives output z_i the value 1, and the absence of a token in place p keeps the device from responding to the value 1 at any other input. When x_i becomes 0 again, transition t_{i+3} fires, and this moves the token in place z_i back to place p. The device is then ready to honor another input request. Because transitions t_1, t_2, and t_3 have a common input place, the modeling rule specifies that only one of these transitions may fire should more than one be enabled at the same time. The model thus specifies that input requests are honored one at a time.



(a) (b)

Figure 2. The 3-input arbiter. (a) Representation. (b) Model.

The model given for the R-S flipflop is very simple and is easy to understand. The same is true for the model of the 3-input arbiter. On the basis of these examples, it would appear that this modeling method is both appropriate and convenient for describing level sequential circuits.

MINIMIZATION

One aspect of our synthesis approach that needs development is a technique for obtaining solutions with a minimum number of secondary variables. Our circuit models are equivalent to primitive flow tables in the classic Huffman synthesis technique, and it is a well known fact that a direct realization of a primitive flow table does not usually give a solution with a minimum number of state variables. For example, if the R-S flipflop were designed in this manner, the solution would have two state variables. It is the state minimization techniques developed for that synthesis approach that allow us to get the solution having just one state variable. With our synthesis approach, a direct realization of a circuit model usually yields a solution with a state variable for each transition. This means that a direct realization of the R-S flipflop model will give a solution with two secondary variables. In this section, we present a first step in the development of a minimization procedure. It is roughly equivalent to the simplest form of row merging in the Huffman synthesis approach.

Our minimization technique is based on the observation that, under certain easily identifiable circumstances, two or more transitions in a model may be replaced by a single transition. When this substitution is made, the modified model has fewer transitions than the original model and will accordingly have fewer state variables. The identification of this reduction possibility and the consequent formation of a modified model are facilitated by the following definitions.

Firing Function: The firing function f_i of transition t_i is a Boolean function having value 1 if and only if t_i is enabled.

Action Set: The action set A_i of transition t_i is the set of all place variables and all output variables that are connected to t_i by a single arc. These are the variables that change value each time t_i fires.

Suppose that transitions t_1 and t_2 in some model have identical action sets. We note that if one or the other of the transitions fires, this causes each variable in the common action set to change value. However, if both transitions fire at the same time, each variable in the action set experiences a double change in value which is the same as no change at all. We can cause identical results with a single transition $t_{1,2}$ defined such that

$$A_{1,2} = A_1 = A_2 \text{ and } f_{1,2} = f_1 \oplus f_2 .$$

Note that in the case where both of the other transitions fire at the same time, the new transition simply does not fire at all. The result is the same. At this point, our reduction technique should be clear. We form the action sets for each transition in our model and then compare them. Wherever two transitions have the same action set, we eliminate them and insert a transition defined as above. For example, examination of the model in Figure 1b shows that

$$A_1 = \{p,y\} \text{ and } f_1 = p\cdot s$$

$$A_2 = \{p,y\} \text{ and } f_2 = y\cdot r\cdot \overline{s}$$

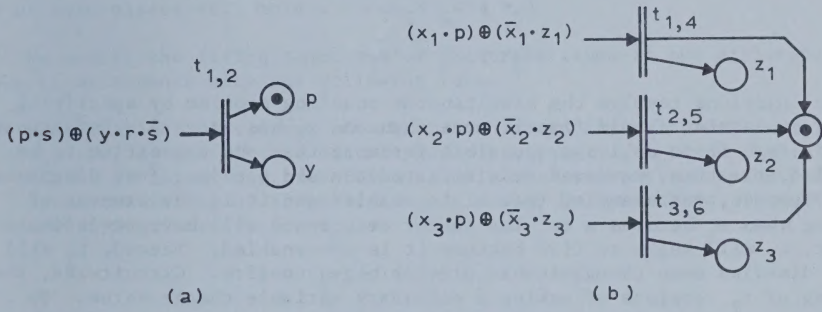
Since both transitions have the same action set, we eliminate them and insert transition $t_{1,2}$ defined such that

$$A_{1,2} = \{p,y\} \text{ and } f_{1,2} = (p\cdot s)\oplus(y\cdot r\cdot \overline{s})$$

The reduced model that results is shown in Figure 3a. The model now has a single transition. All variables in its action set are shown connected to output arcs and it has a single input coming from the expression for its firing function. When the model is synthesized, the input of the transition is treated as an external expression. Direct synthesis of this model gives us the wellknown solution

$$\overset{*}{y} = s + y\cdot \overline{r}$$

If we next examine the model in Figure 2b for reduction possibilities, we find that there are three. The reduced model that results is shown in Figure 3b. Its construction and interpretation are similar to that specified for the model shown in Figure 3a.



$t_{1,2}$
p
$(p\cdot s)\oplus(y\cdot r\cdot\overline{s})$
y
(a)
$t_{1,4}$
$(x_1\cdot p)\oplus(\overline{x}_1\cdot z_1)$
z_1
$t_{2,5}$
$(x_2\cdot p)\oplus(\overline{x}_2\cdot z_2)$
p
z_2
$t_{3,6}$
$(x_3\cdot p)\oplus(\overline{x}_3\cdot z_3)$
z_3
(b)

Figure 3. Reduced models. (a) The R-S flipflop. (b) The 3-input arbiter.

ARBITRATION

In discussing the behavior of the 3-input arbiter, we mentioned a modeling rule to the effect that: If several transitions have a common input place and if more than one of these transitions becomes enabled at the same time, only one of the transitions may fire. In addition, which one fires is not important. This rule was not mentioned in [1] because it referred to a situation that could not be reckoned with at that time. In fact, rule 2 in that paper was meant to indicate that if transitions with a common input place could be enabled at the same time, then the synthesis procedure must not be used. The reason for such a prohibition stems from the fact that circuits generated by the procedure would not have predictable behavior in such circumstances. For example, a single transition might fire or several might fire. The inability of the procedure to handle such situations in an acceptable manner is not a serious drawback. We are dealing here with the most severe of multiple-input-change situations, and circuit design for this type of situation is notoriously difficult.

An obvious thought occurs at this point. A circuit that we have used to illustrate two aspects of our synthesis approach, the 3-input arbiter, cannot be realized by our procedure. In a sense this is true. If one were to directly realize the model in Figure 2b or in Figure 3b, malfunctions would sometimes occur in those situations where the arbitration rule dictates behavior. In this section we present a technique for handling such situations. However, solutions that result from the use of this technique do not have a bounded response time, and as a result, the technique most not be used if a bounded response time is required. In as much as there are applications where the 3-input arbiter is not required to have a bounded response time, we shall illustrate the technique on that device. We begin our discussion by considering a conventional solution to the arbitration problem and showing why it does not work. Then, having developed an understanding of the problem that confronts us, we present the realization technique.

Referring now to the model in Figure 2b, let us eliminate the simultaneous enabling problem by changing the firing functions of t_1, t_2, and t_3 to that shown below.

$$f_1 = p \cdot x_1$$

$$f_2 = p \cdot x_2 \cdot \overline{x}_1$$

$$f_3 = p \cdot x_3 \cdot \overline{x}_1 \cdot \overline{x}_2$$

These functions resolve the simultaneous enabling problem by specifying which transition should fire if more than one x_i has value 1. Unfortunately, even though it is no longer possible for more than one transition to be enabled at a time, we have not eliminated the old problem, just disguised it. Suppose, for example, that t_2 is enabled and is in the process of firing when x_1 becomes a 1. The latter occurrence will have two effects. First, t_1 will begin to fire because it is now enabled. Second, t_2 will become disabled even though it had already begun to fire. Circuitwise, the firing of t_2 consists of making a secondary variable change value. By disabling t_2 just after it had become enabled, we end up having fed the secondary variable a runt input signal, and as a result, the secondary variable may or may not change value. If it does change, then both t_1 and t_2 will have fired, a violation of the arbitration rule.

Having realized this possibility, we next add a delay which will keep t_1 from firing until t_2 has had enough time to stabilize. Then, if t_2 has managed to fire, the token in p will be gone and t_1 will not fire. And, if t_2 has not managed to fire, then t_1 will still be enabled and will fire. The theory here is fine, but it depends on there being a bound on the stabilization time of the secondary variable that realizes t_2, and studies [2] indicate that there is no bound in such situations. This means that there is no delay value which will always work. The larger the value we choose, the more probable it is that the secondary variable has stabilized, but the probability is never 100 percent.

Experience suggests that there is no way of avoiding a stabilization-time problem in a multi-input-change situation of this type. Further, research [2] suggests that there is no solution to the problem if a bounded response time is required of the circuit. However, if the circuit response time is not required to be bounded, then solutions can be achieved by somehow sensing when a state variable is unstable and by using this information to freeze circuit operation until the matter resolves itself. This, in fact, is the basis of the synthesis strategy presented in this section.

Rather than look for a way to keep more than one transition from firing, we propose to modify the transition firing functions in such a way that if an extra firing does occur, the resulting situation will automatically cause corrective action to be taken. Special filter elements will be used to keep the spurious firing and corrective action from being seen by the rest of the circuit and by the circuit outputs.

Suppose that a transition has fired when it should not have. The effect of this firing has been to cause a value change in each variable in the transition's action set. If we could consequently cause the transition to fire a second time, the second firing would cause a second value change which would return the variables to their previous values. This is the corrective action of our strategy.

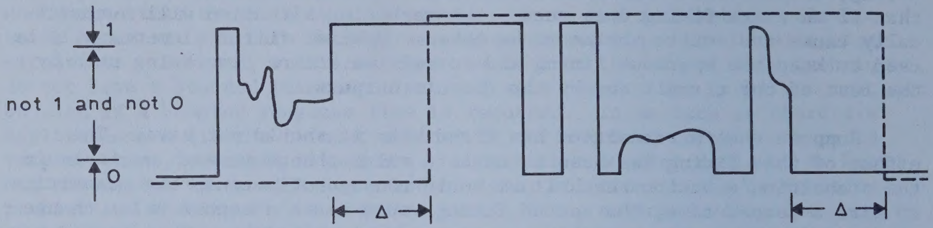
For this strategy to work, we need a way of determining when corrective action needs to be taken. In the case of the 3-input arbiter, detection of extra firings is easy. We know that all places z_i are empty prior to any firing of t_1, t_2, or t_3. Then, if a firing occurs that is consistant with the arbitration rule, two of the places will remain empty and the other will receive a token. But, if more than one transition fires, the problem case, two or more places will hold a token.

We modify the firing functions of the transitions in our arbitration group in accordance with the following rule.

> Modification Rule: Given an arbitration group having transitions t_1, t_2, ..., t_n and given a set of places p_1, p_2, ..., p_n such that prior to the firing of any transition all places are empty and such that after the firing of transition t_i, place p_i holds a token, then, the new firing function for t_i is
>
> $$\text{new } f_i = (\text{old } f_i)\cdot\overline{e}_i + p_i \cdot e_i$$
>
> where e_i is the inclusive OR of all p_j except p_i.

Since the same modification form is used for each transition, we do not attempt to resolve the mutual enabling problem by assigning firing priorities. Further, we do not attempt to keep more than one transition from firing because when all $p_i=0$, all $e_i=0$ and we have the original firing functions. What we do is make the secondary variables that realize the transitions extremely interdependent on one another. This interdependence makes it impossible for these variables to stabilize in a state other than one consistant with the arbitration rule. Any other state will automatically result in more firings. To see how this works, suppose that old f_1 and old f_2 should both become 1 and that transitions t_1 and t_2 should then both manage to fire. This makes e_1 and e_2 become 1 and both transitions consequently experience corrective refiring. If they both manage to refire at about the same time, we return to our initial situation with old f_1 and old f_2 again 1. The transitions will then try to fire again. This will continue until differences in circuit delays result in either $p_1=0$ and $p_2=1$, or $p_1=1$ and $p_2=0$. These are the only conditions for which the circuit will stabilize. Thus, when stabilization does occur, it will do so in a manner consistant with the arbitration rule. Note, however, that the stabilization time cannot be assumed bounded.

We keep the rest of the circuit from seeing these vacillations by using special filter elements. These filters have a single input and a single output, and the output function is illustrated in the figure below.



In this figure, time increases from left to right. The input signal is represented by the solid line and the output signal is represented by the broken line. The figure shows that the device maintains a given output value until the input has had the opposite value for an interval of time Δ. The output then changes value. The notation at the extreme left indicates that the input signal must lie outside of a prescribed middle region in order to be considered a 1 or a 0. This assures that non-standard signals occuring during a vacillation period will not mimic legal values and thereby cause the circuit to appear stable. The thresholds defining the middle region are made appropriate to the logic family used in the circuit implementation. The value chosen for Δ is determined by an analysis of the circuit implementation.

We are now at the point where we can illustrate our procedure for the 3-input arbiter. First, we apply our reduction technique to the model in Figure 2b to get the reduced model shown in Figure 3b. Since all transitions in this model represent transitions from an arbitration group in the original model, we modify their firing functions in accordance with the Modification Rule. This gives

$$f_{1,4} = ((x_1 \cdot p) \oplus (\bar{x}_1 \cdot z_1)) \cdot \overline{(z_2 + z_3)} + z_1 \cdot (z_2 + z_3)$$

$$f_{2,5} = ((x_2 \cdot p) \oplus (\bar{x}_2 \cdot z_2)) \cdot \overline{(z_1 + z_3)} + z_2 \cdot (z_1 + z_3)$$

$$f_{3,6} = ((x_3 \cdot p) \oplus (\bar{x}_3 \cdot z_3)) \cdot \overline{(z_1 + z_2)} + z_3 \cdot (z_1 + z_2)$$

Next, we apply the synthesis procedure given in [1] and get the solution shown in Figure 4. The symbol ⊗ in that figure represents a filter element. The placement of these elements was determined by the fact that variables $y_{i,i+3}$ are realizations of transitions in an arbitration group, and we wish to isolate the device outputs from disturbances occuring during arbitration periods.

The solution is elegantly simple and it is easy to see that it behaves in accordance with our design strategy. Furthermore, the steps involved in obtaining the solution were easy to perform and required very little time. However, the solution is not really complete until a delay value Δ has been determined for each filter element. The method used to determine a delay value is based on the following considerations.

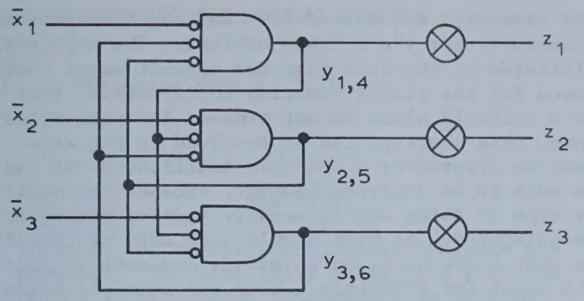


Figure 4. Circuit diagram for the 3-input arbiter.

First of all, we know that a transition is realized by a secondary variable and that the action we call "firing" is realized by a change in value of that secondary variable. Thus, the actions that we are trying to hide are unwanted value changes occurring in a set of secondary variables. Second, the modifications that we make to the firing functions make the values of these secondary variables strongly interdependent. Except for the initial changes that are due to more than one transition being simultaneously enabled, all subsequent changes are due to this interdependence. This interdependence also guarantees that if one secondary variable manages to hold a value different from its pre-disturbance value for a certain length of time, this value will force the other secondary variables to take on their pre-disturbance values and the circuit will be stable.

When a secondary variable y_i takes on a value different from its pre-disturbance value, this causes the condition $p_i=1$ where p_i is the place used to detect the firing of t_i as defined in the Modification Rule. We can thus determine the delay value Δ_i for the filter attached to y_i by determining how long it takes the circuit to become stable given that $p_i=1$. We illustrate this by calculating the delay value $\Delta_{1,4}$ for the filter attached to $y_{1,4}$ in the circuit diagram shown in Figure 4. To begin with, place z_1 served the function of place $p_{1,4}$ in forming this solution, and synthesis showed that $z_1 = y_{1,4}$. Thus, to determine the delay value $\Delta_{1,4}$, we examine the circuit diagram to find out how long it takes to guarantee stability given that $y_{1,4}=1$. First, in order to have this value, we must have $x_1=1$, $y_{2,5}=0$, and $y_{3,6}=0$. Second, we note that this value is forcing $y_{2,5}$ and $y_{3,6}$ to be 0, so that after $y_{1,4}$ has had value 1 for the period of one gate delay, those two secondary variables must have value 0. The circuit is not necessarily stable, though, because the effect of values previously held by those two secondary variables is still propagating through the gate with output $y_{1,4}$. However, if $y_{1,4}$ is still a 1 after the time of two gate delays, the circuit must be stable. Thus, the minimum value for $\Delta_{1,4}$ is the time of two gate delays.

We have now presented our method for handling the arbitration problem and have illustrated it for the 3-input arbiter. The application of the method was facilitated by the fact that our circuit model contained places that could be used for the firing function modification that we require. In models where a suitable place is not present for each transition in an arbitration group, this shortage can be resolved in two ways. First, the needed places can be created by a suitable modification of the model. This approach leaves much to be desired, however, because the modification is a trial and error type of thing and it usually ends up adding more transitions. The second approach is much simpler and adds no transitions. One simply pretends that a needed place p_i exists and uses it wherever necessary in preparing the model for synthesis. When the model is being synthesized, one defines

$$p_i = y_i \oplus dy_i$$

where y_i is the secondary variable that realizes t_i and dy_i is the output of a filter element having input y_i. That filter element will, of course, have to be included in the solution, but y_i would probably have a filter anyway since it realizes a transition in an arbitration group. It is relatively easy to see why this approach works. Prior to the beginning of an arbitration period, $y_i = dy_i$ and so $p_i = 0$. During the arbitration period, changes in the value of y_i cause suitable changes in the value of p_i and the filter output remains unchanged. When the arbitration period ends, if $p_i = 1$, then $y_i \neq dy_i$, and dy_i will soon change value and make $p_i = 0$. However, if $p_i = 0$, then $y_i = dy_i$ and the value of dy_i will not change.

CONCLUSION

We have shown that the synthesis approach presented in [1] applies to level sequential circuits in general and that the application of the approach does not require that circuit behavior be expressed in terms of transition signals. We have also begun the development of a state minimization technique and have given a method for realizing circuits whose behavior is dictated by the arbitration rule. In so doing, we have significantly extended the range of application of the original procedure and have assured the efficiency of its solutions. Most important, we have achieved this without sacrificing any of the features that made the synthesis approach attractive in the first place.

REFERENCES

1. Stucki, M.J., "An Approach for Synthesizing Transition Logic Circuits", Proceedings Eleventh Annual Allerton Conference on Circuit and System Theory, University of Illinois at Urbana-Champaign, (418-427), October 1973.

2. Hurtado, M., "Structure and Performance of Asymptotically Bistable Dynamical Systems", Doctoral Dissertation, Sever Institute of Technology, Washington University, St. Louis, Missouri, May 1975.

SPECIFYING A DESIGN LANGUAGE FOR DIGITAL SYSTEMS

TREVOR MUDGE
University of Illinois, Urbana, Illinois

ABSTRACT

Criteria for specifying a design language are proposed. The similarity these have with the tenets of structured programming is pointed out. A design language is specified using these criteria, which can describe networks of asynchronous logic modules. The process of translating the design language into networks of modules is outlined. An example design is presented. Finally, the design language is shown to satisfy the proposed criteria.

INTRODUCTION

In an attempt to formalize the design process for large digital systems, many researchers have suggested the use of design languages. Examples of such languages and advocacy of their advantages can be found in references 1, 2, 3 and 4. However, using a design language (DL) does not necessarily facilitate the design process. A poorly specified language can encumber the design process and fail to guide it away from design faults. This paper shows that through careful specification a DL can be created for a specific application (in this case the design of networks of asynchronous logic modules) so that by working within the syntax of the language, the designer is forced to formulate his design in a manner that allows him enough freedom to describe any flowchartable process, while at the same time limiting his freedom to describe faulty designs (in this case networks which hang-up). Furthermore this is achieved without the imposition of a complex syntax.

The process of specification is aided by a set of criteria that the DL should satisfy. The underlying motivation is to specify a DL which enables the user to design digital systems efficiently and with the minimum of design errors. Prevention rather than cure is the guideline.

The DL specified translates onto a set of asynchronous logic modules to produce the control structure of a digital system. Actions in the data structure of the system are assumed to be representable as register transfers. The asynchronous modules used by the design language are the W (wye) module, the S (sequence) module, the J (junction) module, the U (union) module, the D (decode) module and the I (iterate) module. These are discussed at length in the literature (references 5, 6, 7) and have been used in paper designs to illustrate their viability in constructing the control structures of complex digital systems (reference 8). The reader is assumed to have some familiarity with these modules.

This work was supported by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-72-C-2059.

A design error, or an ill formed design, is considered to have occurred if the DL describes a network of the above modules, which during the course of normal operation will eventually hang-up or deadlock.

SPECIFYING A DESIGN LANGUAGE

The first step towards specifying a DL is the formulation of a description of the class of designs that the DL is required to produce. The DL should then be specified by a set of syntax rules that satisfy the following three conditions.

C1) The language resulting from the set of syntax rules should include the class of designs that the DL is intended to produce.

C2) The language should not include ill formed designs.

C3) The language should be specified in such a way that the user can easily avoid making syntatical errors.

The requirement that C1 be true of the language is obvious. However it is necessary to check that specifying a DL to satisfy C2 and C3 does not result in it failing to satisfy C1. Given a description of the class of designs that the DL is required to produce and a set of syntax rules, it should be possible to prove whether or not the language resulting from the syntax rules includes the required class of designs. As an example, the DL presented in the next section is required to produce designs which control any flowchartable process, making it suitable for the design of the control structure of a large class of general purpose digital hardware. Examination of the syntax rules shows that such designs are indeed produced within the limits of the syntax rules.

The requirement that C2 be true of the language specified means that a syntax analyser will implicitly check for design faults. If the characteristics of an ill formed design (in our case networks which hang-up) are identified, a set of syntax rules can be formulated which produce designs in the language which are never ill formed. Given a set of conditions which define a well formed design and a set of syntax rules which define the DL it should be possible to prove that the DL includes only well formed designs.

Condition C3 is not a condition that can be shown to have been satisfied by mathematical demonstration as C1 and C2 can. This is because it is a qualitative rather than quantitative condition. A loose characterization might be to say that given two DLs satisfying C1 and C2, the one with the simpler syntax rules more nearly satisfies C2. Whether this makes the user's task easier is open to debate; however, it certainly makes a syntax analyser's task less complex.

The many underlying similarities between the requirements of a good procedure oriented programming language and a DL are highlighted by C1 and C3. These two conditions are inherent in the philosophy of structured programming. In the case of procedure oriented programming languages the class of programs of interest are those with flowchartable control logic. The "Structure Theorem" (reference 9) guarantees that by using the three types of structure f THEN g, IF p THEN f ELSE g and WHILE p DO f (f and g procedures and p a predicate), any flowchartable control logic can be represented. Hence C1 is satisfied for procedure oriented programming languages even when the restricted constructs required for structured programs are used. Whether C3 is satisfied is arguable, as noted above; however, the above three constructs allow the programmer to formulate his programs in a systematic top down fashion (reference 9) which has received wide acceptance as a methodology which is suitable for program formulation by humans, and hence which tends to reduce programmer errors. On the other hand, C2 is not true for procedure oriented programming languages, as certain control logic can be indicated by the programmer which is not executable (ill formed). This is particularly easy if unrestrained use of the GO TO statement is allowed; thus the requirement of structured programming that GO TOs be forbidden, or else used in some very restricted fashion.

A DL is specified in the next section, and at the end of the section it is shown to satisfy the above three criteria.

A DESIGN LANGUAGE

The Syntax

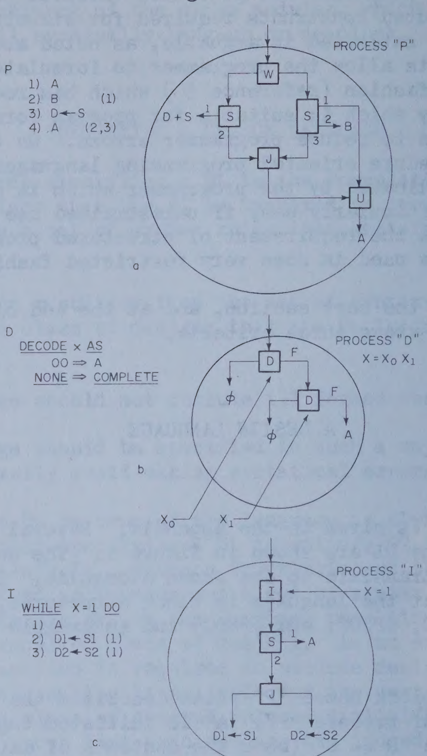
The syntax of the DL is given in the appendix. Several illustrative examples of the use of the DL are shown in figure 1. The networks of modules that these examples translate to are shown alongside. It should be clear from the appendix that the language is block structured. There are three types of blocks (PROC, DPROC, and WPROC) and an example of each is shown in figure 1.

Figure 1(a) shows a block named "P" which describes the following process. Upon the activation of process "P", "A" is initiated together with the register transfer action "D ← S" (move the contents of data cell S to data cell D). When "A" is completed process "B" is to be initiated. When both "D ← S" and "B" are completed process "A" is to be reinitiated. Finally, with the completion of "A", process "P" is considered completed. This is an example of a PROC type block. Notice the integers to the left of the statements. These are used in the parentheses on the right of the statements to indicate sequencing information. For example, statement 4 says that process "A" is to be initiated when the processes indicated in statements 2 and 3 are both completed. No parenthesized integers to the right of a statement indicates it is to be initiated immediately the process described by its block is initiated.

Figure 1(b) shows an example of a DPROC type block. This indicates how the control is to branch according to the state of some external variables (X). The reserved word NONE stands for the union of all those conditions of the external variables not explicitly listed (i.e. $X_0 \vee X_1 = 1$). The reserved word COMPLETE indicates that if NONE is true the empty process (∅) is to be completed. Then the process described by block "D" is considered completed.

Figure 1(c) shows an example of a WPROC type block. In this block process "A" is to be initiated first. When it is completed the register transfer processes "D1 ← S1" and "D2 ← S2" are both to be initiated. When they are both completed the whole block, starting with A, is to be reinitiated as long as the predicate "X=1" holds true. As soon as the predicate is no longer true, and both "D1 ← S1" and "D2 ← S2" have been completed, the process "I" is then considered completed.

Figure 1



```
P
1) A
2) B      (1)
3) D←S
4) A      (2,3)
```

```
D
DECODE × AS
    00 ⇒ A
  NONE ⇒ COMPLETE
```

```
I
WHILE X=1 DO
  1) A
  2) D1←S1 (1)
  3) D2←S2 (1)
```

Space limitations prevent a more complete exposition of the DL. For this the reader is referred to reference 10. However, before closing this subsection on the DL's syntax, a few more comments are in order.

There are only three types of statements in the DL. The IO type, which indicates external asynchronous communications with the control structure. The REG-TRF type, which indicates register transfers in the data structure (e.g. statement 3, figure 1(a)). Finally, the PROC-CALL type, which indicates a process which is described by the block having the same identifier as the PROC-CALL statement (e.g. statement 1, figure 1(a)). Notice the analogy with subroutine calls in procedure oriented programming languages. Natural top down structuring of the design also results from this last type of statement.

The similarities between the DL and a procedure oriented language such as Algol are obvious. Nevertheless there are several important differences. Firstly there is no parameter passing by the PROC-CALL type statements (this would be analogous to subroutine parameter passing). This was done for simplicity and there is no conceptual reason why the DL could not be extended to incorporate this facility. Parameters could be data cells or other blocks, so that a block could be shared by several similar tasks which operate on different data cells. (The translator would have to add switches to the data structure so that different sections of the data structure could be switched to the same section of control structure for processing). Secondly an Algol like language is a sequence of statements whose order of execution is important. In the DL the order in which the statements appear within their blocks, and also the order in which the blocks appear is unimportant. This is because the result of translating a design in the DL onto the modules can be viewed as a directed graph whose nodes are the modules. The position of an arc in the graph is not dependent upon the order in which it is placed into the graph.

The Translation Process

It is appropriate at this point to make some qualitative comments about the translation process. A more in depth quantitative treatment is given in reference 10.

The translation process is quite simple, as can be deduced from figure 1. DPROCs translate to networks of D modules, WPROCs translate to networks of W, S and J modules headed by an I module, and PROCs translate to networks of W, S and J modules. The intra block connections of W, S and J modules are determined by the sequencing or order information given in the parentheses to the right of the statements. The order information can be easily expanded to form a matrix representing the binary relation "is the successor of". This is done below for the block of statements given in figure 1(a). If m(i,j)=1 then the statement labelled i "is the successor of" the statement labelled j.

$$m = \begin{array}{c|ccccc|l} & 0 & 1 & 2 & 3 & 4 & \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & \\ 1 & 1 & 0 & 0 & 0 & 0 & \\ 2 & 0 & 1 & 0 & 0 & 0 & \\ 3 & 1 & 0 & 0 & 0 & 0 & \\ 4 & 0 & 0 & 1 & 1 & 0 & \leftarrow J \\ \hline & \uparrow & & & & & \\ & W & & & & & \end{array}$$

Each statement except 0 and those having no successors (e.g. 4 in the above) translates to a three port S module. Columns with n+1 ones imply n W modules and rows with n+1 ones imply n J modules. A similar relational matrix for inter block connections can be formed and the assignment of U modules deduced.

These matrices, the inter block matrix and the intra block matrices can also be used to complete the syntax analysis. The context free grammar in the appendix which describes the DL's syntax, does not give a complete description of the syntax. Several other constraints must be imposed, so that C2 is satisfied, which cannot be conveniently expressed by a set of production rules. Firstly there are to be no directed cycles in the networks of modules described by the DL. Secondly whenever a process is shared (e.g. A in figure 1(a)), implying the use of a U module, descriptions in the DL which allow the possibility of both input links to the U module to be simultaneously active are to be excluded from the language. (The reason for these constraints are explained more fully in the last subsection of this section). These additional constraints can be checked for by various operations with the relational matrices.

As an example: directed cycles in a block can be identified by forming m, m^2,..., m^{k-1} (k = number of statements in the block) and checking for ones in their main diagonals. Due to the block structure of the DL, the relational matrices are usually small (typically less than 10 X 10), making matrix manipulations quite feasible.

An Example Using The Design Language

The design example presented here is a single instruction computer called SIM (Single Instruction Machine) which performs the single three address instruction:

```
SUBTST A, B, P = A ← C(A) - C(B)
                 IF C(A) = 0 THEN
                 PC ← C(P)   (PC is the program counter)
```

Its control structure is described in the DL below.

Comments:

```
SIM
WHILE RUN = 1 DO
    1) DECI
    2) FETCH  (1)
    3) EXEC   (2)
```

This is the basic instruction fetch and execute cycle. It continues as long as the RUN button is on. Furthermore, it tests the interrupt flag (DECI) before each cycle.

```
DECI
DECODE INT AS
        1 = INTR
        0 = COMPLETE
```

```
INTR
    1) MAR ← IPA
    2) DR ← IP
    3) M ← DR   (1,2)
```

INTR handles the interrupt. IPA holds the address at which the interrupting data is to be stored. IP holds the interrupting data. The interrupt is only for inputting data.

```
FETCH
    1) MAR ← PC
    2) DR ← M       (1)
    3) PC ← INC     (1)
```

This is the FETCH routine. MAR is the memory address register. PC is the program counter. DR is the data register. Note the parallelism between statements 2 and 3.

```
EXEC
     1) IR ← DR
     2) MAR ← IRA      (1)
     3) DR ← M         (2)
     4) OUTPUT DR1     (3)
     5) A ← DR         (3)
     6) MAR ← IRB      (3)
     7) DR ← M         (4,5,6)
     8) DR ← SUB       (7)
     9) MAR ← IRA      (7)
    10) OUTPUT DR2     (8)
    11) M ← DR         (8,9)
    12) DEC            (10,11)
```

This executes the 3 address instruction. IRA holds the address A, IRB holds the address B. OUTPUT DR1 and DR2 are links out of the control structure. These indicate to the external environment that the DR is loaded with data which could be read out to some external device. Since the system is asynchronous, SIM will not continue until the OUTPUT ports receive an acknowledge signal from the external environment.

```
DEC
DECODE DR AS
        0 ... 0 = BR
        NONE = COMPLETE
```

DEC checks to see if the result of the subtraction is zero.

```
BR
    1) DR ← PC
    2) PC ← IRP      (1)
    3) MAR ← PC      (2)
    4) M ← DR        (3)
    5) PC ← INC      (3)
```

If the result is zero, BR handles the branch of control. C(PC) is loaded at the branch address and PC is loaded with the branch address.

The resulting data structure is shown in figure 2. Symbols such as I represent identity operators. The one indicated moves data from the program counter to the memory address register when requested to do so by the S module corresponding to the DL register transfer statement "MAR ← PC" (see FETCH block statement 1).

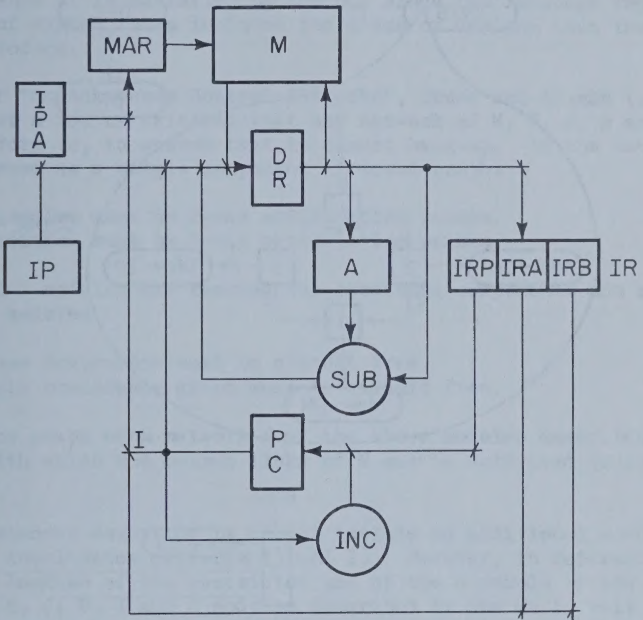


Figure 2

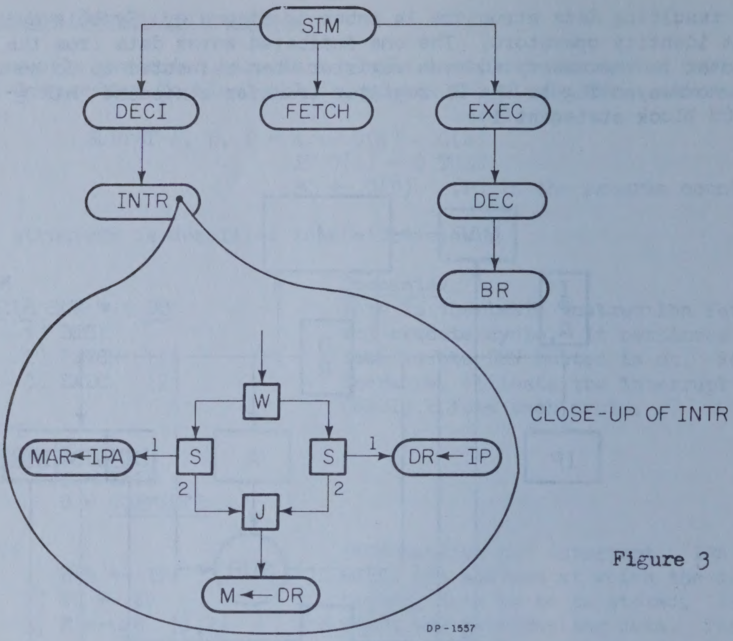
When a control structure is hooked up to a particular data structure the determinism of the total system must be verified. The presentation of a systematic method to check for determinism is outside the scope of this paper. In the above example the determinism was checked by visual examination of the register transfer statements in the DL description. (This would clearly be inadequate in a much larger design).

The block structure of the design is shown in figure 3, together with a close-up of the inside of one of the block (INTR). Figure 4 shows the realization of the most complex block (EXEC) in terms of the modules.

The Design Language And The Three Specification Conditions

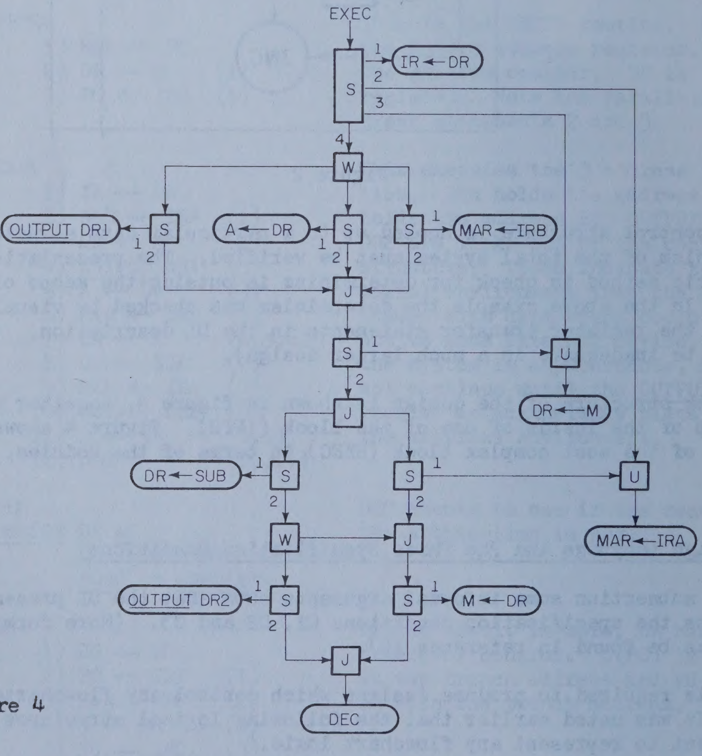
In this subsection some informal arguments show that the DL presented above satisfies the specification conditions C1, C2 and C3. (More formal arguments can be found in reference 10).

The DL is required to produce designs which control any flowchartable process. It was noted earlier that the following logical structures were sufficient to represent any flowchart logic.



SIM
DECI
FETCH
EXEC
INTR
DEC
BR
W
S
S
MAR←IPA
DR←IP
1
1
2
2
J
M←DR
CLOSE-UP OF INTR
DP-1557

Figure 3



EXEC
S
1
IR←DR
2
3
4
W
OUTPUT DR1
S
A←DR
S
S
MAR←IRB
1
2
1
2
1
2
J
S
1
U
2
DR←M
J
DR←SUB
S
1
S
1
U
2
2
W
J
MAR←IRA
OUTPUT DR2
S
1
S
1
M←DR
2
2
J
DEC

Figure 4

1) f THEN g
2) IF p THEN f ELSE g
3) WHILE p DO f

These three logical structures exist in the DL; 3) exists explicitly - the WHILE block (WPROC), 2) exists in a more general form - the DECODE block (DPROC), and 1) exists in a more general form - the sequencing or order information. Hence C1 is satisfied by the DL, since the language resulting from the set of syntax rules includes the class of designs that the DL is required to produce.

In their paper "Asynchronous Control Networks", Bruno and Altman (see reference 6) present a set of criteria that any network of W, S, J, D and I modules must conform to, to ensure that it cannot hang-up. If the network of modules is viewed as a single component directed graph:

1) I modules must be 2-way articulation points.
2) D modules must be 3-way articulation points.

When the I and D modules are removed the remaining components are composed of W, S and J modules.

3) These components must be circuit free.
4) Their precedence graph must be circuit free.

(The precedence graph of a network N of the above modules describes the relative order with which the output links of N can be activated (references 6 and 7)).

The control networks described by the DL include an additional module, the U module. This invalidates criteria 1) and 2). However, in reference 10 it is shown that because of the restricted use of the U module by the DL, any network of W, S, J, D, I and U modules described by the DL is well formed provided an associated network of W, S, J, D and I modules satisfies the above criteria. It is also shown that if the design in the DL is syntactically correct, the separate blocks for I and D modules (WPROCs and DPROCs) together with the fact that entry to a block is always thru a single link, imply criteria 1), 2) and 4) are satisfied by the associated network. Lastly, it is shown that checking for 1), 2) and 4) in a directed graph corresponds to recognizing the context free language which is the DL, a computationally much more efficient procedure. Criterion 3) is satisfied by a syntactically correct design, as is the additional constraint, the U condition. (The U condition requires that both input links to a U module never be active simultaneously). It was noted in the subsection on the translation process, that these two remaining conditions were checked for by the manipulation of small matrices. Hence the DL satisfies C2, since it does not include ill formed designs.

Whether the DL satisfies C3 is a matter of opinion. The following closing remarks of this section give the author's opinion. The block structuring, especially the requirement of separate blocks for WHILE and DECODE statements should help the designer to formulate his design in an error free fashion. Furthermore, the top down structuring of the blocks induced by the PROC-CALL type statements maintains a correspondence between the textual description of the digital machine and its proposed operation that should also facilitate the design process. The only sources of syntactical error not made conspicuous by the form of the syntax are U condition conflicts and the creation of directed circuits thru incorrect use of the order information.

APPENDIX

The syntax is given in extended BNF. Use is made of the following meta-symbols.

```
   A[B]  ⇔  A|AB
 A{B|C}  ⇔  AB|AC
   {A}+  ⇔  A|AA|AAA|...
   {A}*   =  {A}+|∅
```

Non-terminal symbols are sequences of uppercase letters and hyphens. They are separated by blanks. The terminal symbols are underlined. Those that are spelt differently in the text are listed together with their spelling below.

DL	carriage return, line feed	INITIATE	➔
LP	(	ASSIGN	→
RP	)	COMMA	,
LETTER	A,...,Z	ZERO	0
DIGIT	1,...,9	ONE	1

Blanks may be inserted between terminals for ease of reading.

```
PROGRAM ::= {PROC|DPROC|WPROC}+

PROC  ::= DL PROC-ID STAT-LIST
DPROC ::= DL PROC-ID DSTAT
WPROC ::= DL PROC-ID WSTAT STAT-LIST

STAT-LIST ::= {DL LABEL STAT}+
STAT      ::= {PROC-CALL|REG-TRF|IO} ORDER-INFO

PROC-CALL ::= PROC-ID
REG-TRF   ::= NAME ASSIGN NAME
IO        ::= {INPUT|OUTPUT} PROC-ID

DSTAT ::= DL DECODE NAME AS DLIST
DLIST ::= {DL BITS INITIATE ACTION}*DL NBITS INITIATE ACTION

WSTAT ::= DL WHILE PRED DO
PRED  ::= NAME REL {NUM|NAME}

PROC-ID ::= ID
NAME    ::= ID SUB-OPTION
ID      ::= LETTER{LETTER|DIGIT}*
LABEL   ::= NUM RP
NUM     ::= {DIGIT}+

ORDER-INFO ::= SUB-OPTION
SUB-OPTION ::= [LP SUB-LIST RP]
SUB-LIST   ::= NUM{COMMA NUM}*

ACTION ::= COMPLETE|PROC-ID
NBITS  ::= BITS|NONE
BITS   ::= {ZERO|ONE}+

REL ::= useful binary relations
```

REFERENCES

1) Computer Magazine. IEEE Computer Soc., Dec. 1974.

2) Chu, Y. "An ALGOL Like Computer Design Language," Comm. ACM (Oct. 1965), vol. 8, pp. 607-615.

3) Duley, J.R.; and Dietmeyer, D.L. "A Digital System Design Language (DDL)," IEEE Trans. Computers (Sept. 1968), vol. C-17, no. 9, pp. 850-861.

4) Friedman, T.D. "ALERT: A Program to Compile Logic Designs of New Computers," Digest 1st Annual IEEE Comp. Conf. (Sept. 1967), pp. 128-130.

5) Patil, S.S.; and Dennis, J.B. "The Description and Realization of Digital Systems," IEEE Comp. Conf. (Sept. 1972), pp. 223-226.

6) Bruno, J.; and Altman, S. "Asynchronous Control Networks," IEEE Conf. Proceeding 10th Annual Symposium on Switching and Automata Theory (1969), pp. 61-73.

7) Altman, S.M.; and Denning, P.J. "Decomposition of Control Networks," Record of the Proj. MAC Conf. on Concurrent Systems and Parallel Computation (1970), pp. 81-92.

8) Dennis, J.B. "Modular, Asynchronous Control Structures for a High Performance Processor," Record of the Proj. MAC Conf. on Concurrent Systems and Parallel Computation (1970), pp. 55-80.

9) Mills, H.D. Mathematical Foundations of Structured Programming. IBM, 1972.

10) Mudge, T. A Design Language for Modular Asynchronous Control Structures. CSL Report No. R- , Univ. of Illinois, to be published.

A HEURISTIC GUIDED ALGORITHM FOR OPTIMAL BACKBOARD ORDERING

ALBERTO SANGIOVANNI-VINCENTELLI*
Dept. of Electrical Engineering
and Computer Sciences
University of California, Berkeley

MAURO SANTOMAURO
Istituto di Elettrotecnica
ed Elettronica
Politecnico di Milano,
Milano - Italy

ABSTRACT

Heuristically guided search is a method developed in Artificial Intelligence by Hart, Nilsson and Raphael for using heuristic information from the problem domain to direct the search of an optimal solution of a problem represented in a particular way, saving computation time and storage requirement.

In this paper, the HRN method is used to solve a layout problem: the optimal placement of mother boards of a large scale complex electronic circuit.

I. INTRODUCTION

It is well known that the design of almost every large system requires as one of the last stages the solution of layout problems. In particular, the placement and the routing of elements are essential problems to be solved in complex electronic circuit design [1,2,3]. One of the most important placement problem is the optimal placement of mother boards carrying the semiconductor chips. It is assumed that the mother boards have to be placed on a straight line and that they have to be equally spaced. A criterion which can be followed in the placement of these elements is the minimization of the length of interconnecting links [3]. An efficient algorithm for finding an optimal placement is not available at the moment, due to the intrinsic complexity of this problem. Some attempts have been made in [3]. In this paper, a method developed in problem solving [4,5] a branch of Artificial Intelligence, is proposed in order to lower the computational complexity. In particular, the state space approach to problem solving is used in order to represent the optimal placement problem, and an algorithm, which can obtain an optimal solution, based on heuristic information from the problem domain, is introduced and its complexity is discussed.

II. STATE SPACE REPRESENTATION OF OPTIMAL BACKBOARD ORDERING

The goal of problem solving is to devise efficient methods for automatically obtaining the solution of a problem proposed to a computer in an appropriate way [4,5]. One of the most important approaches to problem solving is the State Space (SS) approach [4,5]. In order to obtain a description of a given problem in SS, some particular elements have to be individuated:

(i) the problem "configurations;" an element v

called state is associated to each configuration

(ii) the initial and the final configurations

---

[*] On leave from Istituto di Elettrotecnica ed Elettronica, Politecnico di Milano - Italy.

(iii) *the "transformations"* which change a problem configuration into another one; a function $\gamma : s_i \rightarrow s_j$, called *operator* is associated to each transformation.

A *solution* of the SS problem is a sequence of operators which lead from the initial state to a final state. A *cost* $c(\gamma)$ is often associated to each operator; then, an *optimal solution* can be searched, i.e., a solution such that the sum of the costs associated to its operators is minimum.

It is possible to represent a SS with a directed graph $G = (V,A)$. The nodes of G, V, are corresponding to the states of the problem; an arc $a_{ij} = (v_i,v_j) \in A$ if there exists an operator γ_h s.t. $\gamma_h(v_i) = v_j$. v_j is said to be a *successor* of v_i and v_i is called *parent* of v_j. A weight $c(v_i,v_j)$ is attached to each arc a_{ij} and is equal to $c(\gamma_h)$. Then an optimal solution of the problem is equivalent to a shortest path in G.

The optimal backboard placement problem admits a straightforward SS representation. In particular, let $N = \{n_1,\ldots,n_r\}$ be the set of all the mother boards to be placed and $P(N)$ the power set of N. Let $S \subseteq P(N)$ be the set of all the linkages among the mother boards (usually given via a connection matrix). Let $E(\tilde{N}_k)$, $\tilde{N}_k \in P(N)$, be the set of the linkages which join the mother boards in $\tilde{N}_k$ with all the other mother boards:

$$E(\tilde{N}_k) = \{s_j \in S \mid s_j \cap \tilde{N}_k \neq \phi \quad s_j \cap (N-\tilde{N}_k) \neq \phi\}$$

A correspondence between the elements in $P(N)$ and the states of the problem can be stated. The initial state is ϕ and the final state is N. An operator $\gamma(v_i,v_k)$ joins two states v_i and v_k iff the corresponding sets $\tilde{N}_i$ and $\tilde{N}_k$ are such that $\tilde{N}_k = \tilde{N}_i \cup \{n\}$. In other words an operator "adds" a mother board to a set of mother boards. The cost $c(v_i,v_k)$ associated to an operator is given by $e(\tilde{N}_k) = |E(\tilde{N}_k)|$.

It can be easily proved that an optimal solution of the SS problem individuates an optimal ordering of the mother boards with respect to the total length of the interconnecting links. If the graph representation of the SS problem is taken into account, the optimal placement problem is corresponding to a shortest path in $G(V,A)$ [3].

III. THE HEURISTIC GUIDED ALGORITHM

Since the dimensions of the SS for real problems is usually of big magnitude, it is mandatory to devise some procedures which do not involve the search of the whole graph. In fact, it has to be underlined that the graph does not need to be stored *explicitly*, i.e., with the complete list of all the vertices and of all the arcs, but it can be described *implicitly* i.e., with the initial state, the final state(s) and the "rules" which generate the successors of the initial node step by step as long as the final state is reached.

Therefore, it has to be noted that an optimal solution search strategy consists of two processes:

(i) incremental construction of the path

(ii) incremental explicitation of the graph called *expansion technique*

The goal of the algorithms developed in problem solving is to expand a part, as small as possible, of the graph without loosing the possibility of finding an optimal solution. In fact, it has be noted that the smaller the expanded part of the graph is, the lower the computational effort and the storage requirements are. The general structure of the search algorithms is as follows: at each step, a node of the subgraph of G already generated is selected, its successors are generated (this process is called *expansion* of the node) and pointers are set up from each successor back to its parent node. The nodes are selected at each step according to an *evaluation function* $\hat{f}(v)$. In particular, Hart, Raphael, Nilsson (HRN) or Heuristic Guided Algorithm [4,5] uses informations about the problem represented by the graphs, called *heuristic informations*, and codes them in $\hat{f}(v)$, which is an estimate of the cost of a minimal cost path from i to a goal node constrained to go through node v. This function $\hat{f}(v)$ can be expressed as

$$\hat{f}(v) = \hat{g}(v) + \hat{h}(v)$$

where $\hat{g}(v)$ is an estimate of the cost $g(v)$ of a minimal cost path from i to v and $\hat{h}(v)$ is an estimate of the cost $h(v)$ of a minimal cost path from v to a goal node. Here, $\hat{g}(v)$ is chosen as the cost of the path from i to v given by summing the arc costs encountered while following the pointers from v to i (this path is the lowest cost path from i to v found so far) and $\hat{h}(v)$ can be obtained using the heuristic information available from the problem domain.

The algorithm consists of the following sequence of steps:

(1) Put the start node i on a list called OPEN. Set $\hat{g}(i) = 0$ and $\hat{f}(i) = 0$

(2) If OPEN is empty exit with failure; otherwise continue.

(3) Remove from OPEN that node v whose f value is smallest and put it on a list called CLOSED. (Resolve ties for minimal f values arbitrarily, but always in favor of any goal node).

(4) If v is a goal node, exit with the solution path obtained by tracing back through the pointers; otherwise continue.

(5) Expand node v, generating all of its successors. (If there are no successors, go to (2)). For each successor v_i, compute

$$g_i = \hat{g}(v) + c(v,v_i).$$

(6) If a successor v_i is not already on either OPEN or CLOSED, set

$$\hat{g}(v_i) = g_i \quad \text{and}$$

$$\hat{f}(v_i) = g_i + \hat{h}(v_i).$$

Put v_i on OPEN and direct a pointer from it back to v.

(7) If a successor v_i is already on OPEN or CLOSED and if $g(v_i) > g_i$, then update it by setting

$$\hat{g}(v_i) = g_i \quad \text{and}$$

$$\hat{f}(v_i) = g_i + \hat{h}(v_i).$$

Put v_i on OPEN if it was on CLOSED and redirect to v the pointer from v_i.

(8) Go to (2).

It is possible to prove that, if $\hat{h}(v)$ is a lower bound on the cost $h(v)$ of a minimal cost path from node v to a goal node, then HRN algorithm is admissible, i.e. it always finds an optimal path. Moreover, if we make a further restriction on the estimate $\hat{h}$, it is also possible to prove that an HRN algorithm A* is optimal in the following sense: Algorithm A* never expands more nodes than does any other admissible algorithm A such that A* is more informed than A. We say that an algorithm A* is more informed than A when the estimate h(v) used by algorithm A* is strictly larger (for each nongoal node v) than the estimate used by algorithm A.

The above result is true if we make the following assumption: For any two nodes v_i and v_j which are connected by an arc, we have

$$\hat{h}(v_i) - \hat{h}(v_j) \leq c(v_i, v_j).$$

This assumption is called the consistency assumption and its meaning is that, by moving from a node to any successor, we always have a better estimate.

With the consistency assumption it is possible to prove that, when A* expands a node v, it has already found an optimal path to v. Therefore, algorithm A* will never reopen a closed node. In many cases, however, the estimate does not satisfy the consistency assumption and, thus, the above results on the optimality of A* are no longer true.

In order to apply HRN to the optimal backboard ordering, it is necessary to introduce an "estimate" which satisfies the "lower bound" condition.

The proposed estimate is

$$h_o(v_k) = \frac{\sum\limits_{n_i \in N-\tilde{N}_k} e(n_i) - \min\left(\max\limits_{n_i \in N-\tilde{N}_k} e(n_i), e(\tilde{N}_k)\right)}{2} \qquad (3.1)$$

In the following Theorem, the consistency of h_o will be proved.

Theorem 1. h_o is consistent, i.e.,

$$\forall v_j, v_k \in V : h_o(v_i) - h_o(v_k) \leq c(v_j, v_k) \qquad (3.2)$$

Proof. Case A: $e(\tilde{N}_j) \geq \max\limits_{n_i \in N-\tilde{N}_j} e(n_i)$

Being $\tilde{N}_k = \tilde{N}_j \cup \{n\}$, (3.2) can be written as follows:

$$e(n) - \max_{n_i \in N-\tilde{N}_j} e(n_i) + \min\left(\max_{n_i \in N-\tilde{N}_k} e(n_i), e(\tilde{N}_k)\right) \leq 2e(\tilde{N}_k) \qquad (3.3)$$

Because obviously $e(\tilde{N}_k) \geq \min\left(\max\limits_{n_i \in N-\tilde{N}_k} e(n_i), e(\tilde{N}_k)\right)$, (3.3) can be rewritten as follows:

$$e(n) - \max_{n_i \in N-\tilde{N}_j} e(n_i) \leq e(\tilde{N}_k) \qquad (3.4)$$

As $n \in N-\tilde{N}_j$, the right hand side of (3.4) is always less or equal to 0. Being $e(N_k) \geq 0$, the inequality is always satisfied.

Case B: $e(\tilde{N}_j) \leq \max_{n_i \in N-\tilde{N}_j} e(n_i)$

By the same reasonings, developed in Case A, (3.2) can be expressed as follows:

$$e(n) - e(\tilde{N}_j) \leq e(\tilde{N}_k) \tag{3.5}$$

Moreover, the following equality holds:

$$e(\tilde{N}_k) = e(\tilde{N}_j) - |E(\tilde{N}_j) \cap E(n)| + |E(n) - (E(n) \cap E(\tilde{N}_j))| \tag{3.6}$$

or

$$e(\tilde{N}_k) = e(\tilde{N}_j) - 2|E(\tilde{N}_j) \cap E(n)| + e(n) \tag{3.7}$$

Therefore (3.5) becomes:

$$-e(\tilde{N}_j) \leq e(\tilde{N}_j) - 2|E(\tilde{N}_j) \cap E(n)| \tag{3.8}$$

which is, obviously, always satisfied.

It has to be pointed out that HRN algorithm is essentially a branch-and-bound technique. Therefore, the whole graph of the problem may be expanded in the worst case. The algorithm proposed by Cederbaum [3] always needs the previous expansion of the graph and, after the expansion, a shortest path in the graph is computed via a min-sum algorithm. It has to be noted that the graph of the problem has 2^r nodes where r is the number of mother boards to be placed. For this reason, Cederbaum's algorithm is not suited for large scale problems. On the other hand, HRN algorithm has been successfully applied to difficult large scale combinatorial problems like the travelling salesman problem [6]. Moreover, it is possible to insert in HRN algorithm some simple steps which aim to limit the storage requirements during the execution of the algorithm itself. In this case, the optimal solution cannot be guaranteed, but anyway nearly optimal solutions are achieved. Obviously the power of HRN algorithm has to be evaluated "in the mean" carrying on tests on many randomly generated problems; for this reason a program based on HRN algorithm has been devised by the authors and is being tested vs. Cederbaum's algorithm in order to give a precise figure of the achieved gain.

IV. CONCLUDING REMARKS

In this paper a particular layout problem has been considered: the optimal placement of mother boards carrying the semiconductor chips. The optimization criterion followed is the minimization of the total length of the links between the mother boards.

An optimization technique devised an Artificial Intelligence has been introduced and applied to this layout problem. The proposed algorithm is basically a shortest path algorithm on a graph not explicitly given with the aim of reducing the expanded part of the graph during the shortest path search.

The capability of limiting the search is due to a particular evaluation function associated with the nodes of the graph called estimate or heuristic function. A heuristic function has been proposed and its properties have been proved. Some computational remarks have been finally given.

Future research directions can be indicated as:

(i) study of new, more informed, heuristic functions

(ii) devise of heuristic (not optimal) algorithms which avoid the back track procedure

(iii) theoretical and experimental comparison between HRN algorithm and heuristic algorithms.

These points are being investigated by the authors; in particular, a more informed estimate has been very recently found and is presently being tested on the UNIVAC 1108 of the Politecnico di Milano.

REFERENCES

[1] M.A. Breuer, "General Survey of Design Automation of Digital Computers" Proc. IEEE, vol. 54, pp. 1708-1721, Dec. 1966.

[2] L. Steinberg, "The Backboard Wiring Problem. A Placement Algorithm," SIAM Rev., vol. 3, pp. 37-50, Jan. 1961.

[3] I. Cederbaum, "Optimal Backboard Ordering Through the Shortest Path Algorithm." IEEE Trans. on Circuits and Systems, vol. CAS-21, pp. 626-632, Sept. 1974.

[4] N. Nilsson, "Problem Solving Methods in Artificial Intelligence," McGraw Hill Co., 1971.

[5] A Sangiovanni Vincentelli, M. Somalvico, "State Space Approach in Problem-Solving Optimization," 5th Conference on Optimization Techniques, Springer-Verlag, 1973.

[6] I. Pohl and P. Banwell, "A Heuristic Program for the Traveling Salesman Problem," Memo K-3, Univ. of Calif., Santa Cruz, 1973.

INDEX OF AUTHORS

Adrion, W. R. 185

Anderson, J. B. 2, 530

Austin, J. W. 103

Bandler, J. W. 184, 633, 643

Barak, A. 867

Bertsekas, D. P. 192, 622

Bien, Z. 809

Bose, N. K. 251

Bredeson, J. G. 682

Brown, D. P. 382

Bucci, G. 104

Burris, F. E. 485

Butz, A. R. 505

Calo, S. 65

Carter, C. R. 477

Chan, H. C. 477

Charalambous, C. 184

Chen, J. H. K. 184, 633

Chevillat, P. R. 1

Chow, J. H. 48

Chu, W. Y. 184

Chyung, D. H. 809

Cirjanic, R. B. 2

Conan, J. 612

Costello, D. J., Jr. 1, 28

Cruickshanks, A. 859

Curtis, R. A. 590

Dalsgaard, P. 633

De Santis, R. M. 612

Dickinson, B. W. 356

Dines, P. 515

Druffel, L. E. 450

Dussault, J. 716

Elahi-Taleghani, A. 28

Elliott, D. L. 605

Eslami, M. 382

Eyman, E. D. 168, 174

Filicori, F. 663

Fishman, P. M. 240

Franks, L. E. 756

Frick, P. A. 85

Fu, K. S. 114

Funahashi, Y. 363

Galiana, F. D. 397

Gitlin, R. D. 781

Goto, S. 431

Guaccimanni, L. A. 104

Haddad, A. H. 227

Hakimi, S. L. 440

Halski, D. J. 848

Hartman, J. 523

Hazony, D. 258, 515

Hemami, H. 124

Hickin, J. 39
Hill, F. S., Jr. 764
Hurtado, M. 605
Hutton, M. F. 29
Huynh, T. H. 792
Ishak, W. S. 59
Jackson, G. L. 284
Jamshidi, M. 813
Jaswa, V. C. 124
Javid, S. H. 839
Jenkins, L. 598
Johannes, J. D. 450
Jong, M. T. 58
Kaliski, M. E. 388
Kajitani, Y. 269
Karni, S. 297
Kassam, S. A. 204
Klein, Q. L. 388
Kodandapani, K. L. 747
Krieger, M. 716
Krishman, K. R. 859
Kuszta, B. 829
Lal, M. 261
Latino, C. D. 682
Law, C.-W. 2
Lecours, M. 792
Lee, T. N. 590
Leighton, W. J. 11
Leiner, B. M. 335
Linn, J. K. 297
Liu, P. C. 633
Liu, R. 598
Lugannani, R. 241
Luh, J. Y. S. 141
McAfee, L. C., Jr. 653
McGhee, R. B. 124
McLane, P. J. 782
Malik, N. R. 284
Maruyama, T. 269
Massey, J. L. 539
Mayeda, W. 414, 873
Meditch, J. S. 801
Metze, G. 692, 704, 737
Metzner, J. J. 75
Meyer, G. G. L. 93
Michel, A. N. 569, 580
Miller, M. H. 672, 681
Miller, R. K. 569, 580
Milstein, L. B. 21
Modarressi, A. R. 251
Modestino, J. W. 211
Monaco, V. A. 663
Mudge, T. 905
Neuhoff, D. L. 325
Newman, C. M. 234
Northouse, R. A. 158
Nussbaum, H. 345
Ohtsuki, T. 431

Oppenheim, A. V. 457
Ozguner, F. 737
Papantoni-Kazakos, P. 220
Patrinos, A. N. 440
Pepper, M. L. 887
Persoon, E. 114
Priemer, R. 467
Razzaghi, M. 813
Rhodes, I. B. 848
Rice, S. O. 241
Rizk, M. R. M. 643
Roman, J. R. 373
Rubin, I. 345, 523
Rubinstein, J. E. 421
Sangiovanni-Vincentelli, A. 407, 916
Santomauro, M. 916

Schmidt, D. C. 450
Schwartz, S. 65
Sekhon, H. S. 495
Seth, S. C. 747
Shanmugam, K. S. 58, 261
Shea, P. D. 103
Shinoda, S. 269, 414
Shneydor, N. 560
Singer, E. 457
Sinha, N. K. 39, 59, 829
Sinha, V. P. 495
Smith, J. E. 692, 704
Snir, M. 867
Snyder, D. L. 240
Stamm, J. C. 467
Stuck, B. W. 234
Stucki, M. J. 887, 896
Subrahmanyam, M. D. 168
Sudhakar, D. P. 174
Szulewski, P. A. 85
Tan, H. H. 11, 307
Thomas, J. B. 204
Thorbjornsen, A. R. 277
Tromp, H. 643
Uhran, J. J. 539
Vandelinde, V. D. 774
Varshney, P. K. 227
Waltz, D. L. 151
Wojcik, A. S. 727
Wolfe, C. A. 801
Wu, W. W. 548
Yam, E. S. 141
Yao, K. 314
Yoshimura, T. 431
Yue, O. 241
Zames, G. 560

OUND BY

UNIVERSITY OF ILLINOIS-URBANA
621.31921AL54P C001
PROCEEDINGS.$URBANA ILL
13 1975
3 0112 008033638